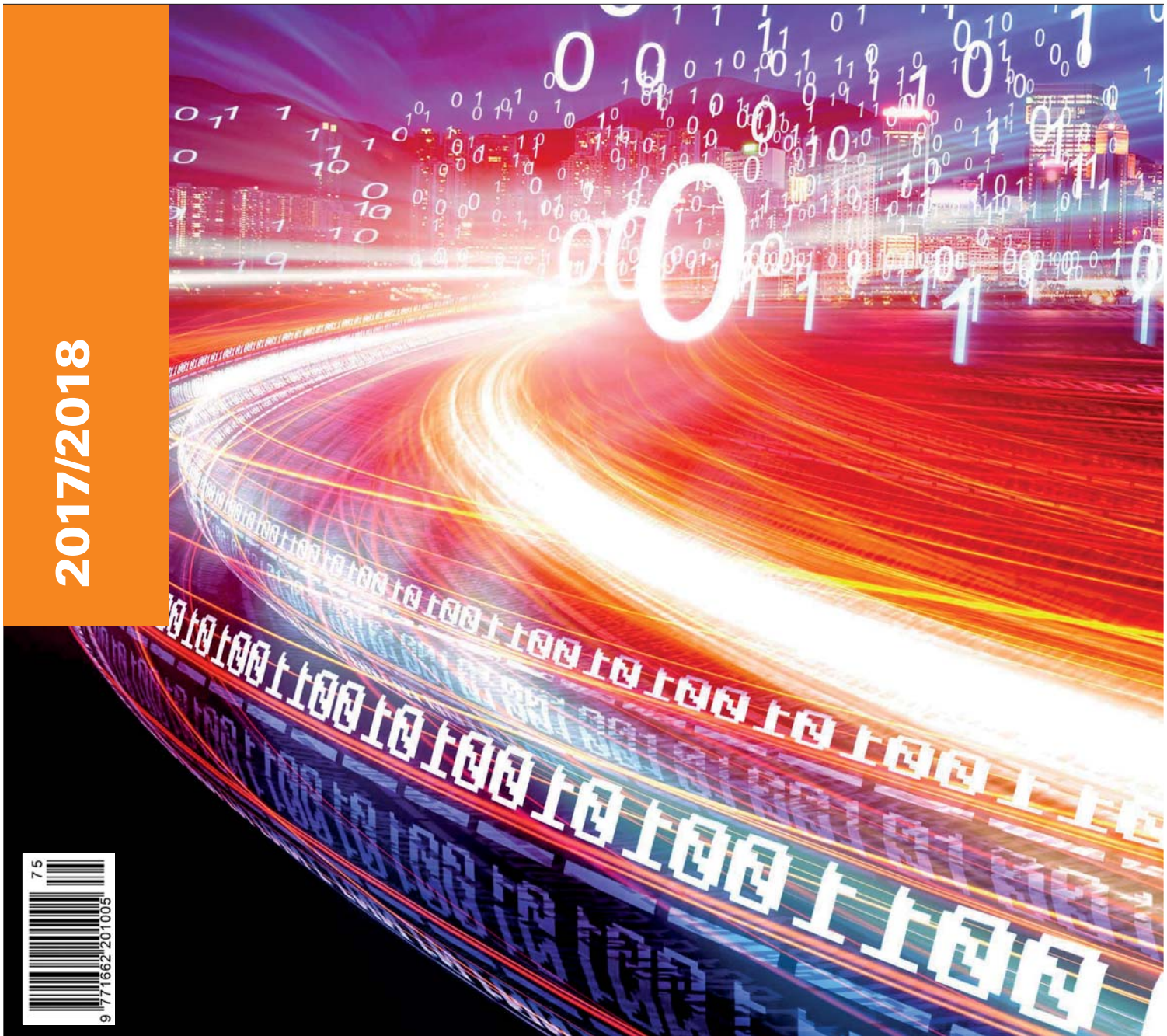
Fachhochschule Nordwestschweiz
Hochschule für Technik

Institut für Mobile und Verteilte Systeme

FOKUS REPORT

2017/2018



9 771662 201005 75

Fast and Smart

Weltweit werden mehr als 40000 Abfragen bei Google pro Sekunde abgesetzt[1]. Jede dieser Abfragen generiert einen Schwall von Daten: einerseits die Suchergebnisse, die über das Internet zum abfragenden Rechner übermittelt werden, anderseits aber die Abfrage selber, welche bei Google kontextbasiert analysiert und in einer aggregierten Form gespeichert wird. In der gleichen Sekunde werden auch 300 Minuten Video bei YouTube hochgeladen und 72000 Videos konsumiert, 2.6 Millionen E-Mails übermittelt, 6000 Tweets bei Twitter versandt und 820 Bilder auf Instagram veröffentlicht. Alles in allem werden 52 TByte Daten pro Sekunde durch das Internet geschleust und dabei 38 MWh Strom verbraucht, d.h. es werden über 110 Kernkraftwerke von der Grösse Leibstadt alleine fürs Internet benötigt.

Gemäss Schätzungen der *International Data Corporation* (IDC)[2] aus dem Jahr 2014 wird im Jahr 2020 jede der 4.2 Milliarden Personen mit Internetanschluss durchschnittlich 1.7 MByte Daten pro Sekunde produzieren oder durch Computer und andere Geräte produzieren lassen und zu den bislang gespeicherten 44000000000000000000000 (44 Zetta) Bytes hinzufügen. Glücklicherweise ist die Datenaufzeichnungsdichte von Festplatten dermassen hoch, dass wir diesen astronomisch grossen Datenberg optisch kaum wahrnehmen. Um trotzdem ein Bild vor Augen zu haben, könnten wir uns eine herkömmliche Festplatte als sehr dünnes Haar von der Erde bis zur Sonne vorstellen, welche immerhin schon bis zur Hälfte vollgeschrieben wäre. Die Studie von IDC weist auch darauf hin, dass nur ca. 20% aller Daten durch spezielle Massnahmen geschützt sind, obwohl 35% der Daten einen solchen Schutz benötigen würden. Nur etwa 5% aller Daten sind in maschinenlesbarer Form annotiert und lediglich ein Sechstel davon wird für Analysezwecke genutzt. Selbstverständlich möchten verschiedene Unternehmen, die sich auf das «Goldschürfen» in Datenbeständen spezialisiert haben, einen vielen grösseren Anteil annotieren und analysieren.

Neben den Internet-Usern generieren vor allem Computer, Sensoren und andere Geräte, welche im *Internet of Things* (IoT) subsumiert werden, den Grossteil aller erzeugten Daten. Je nach Betrachtungsweise handelt es sich bei diesem angesammelten riesigen Datenberg um eine Goldgrube, einen Müllhaufen oder wahrscheinlich beides gleichzeitig. Teilen wir den zukünftigen Datenberg von 2020 gleichmässig auf die 4.2 Mia Internetbenutzer auf, so würden bei jedem von uns

1 http://www.internetlivestats.com
2 https://www.emc.com/leadership/digital-universe/2014iview/index.htm

Inhalt

Impressum

Herausgeberin:
Fachhochschule Nordwestschweiz FHNW
Institut für Mobile und Verteilte Systeme
Bahnhofstrasse 6
CH-5210 Brugg-Windisch
www.fhnw.ch/technik/imvs
Tel +41 56 202 99 33

Kontakt: Prof. Dr. Jürg Luthiger
juerg.luthiger@fhnw.ch
Tel +41 56 202 78 23
Fax +41 56 462 44 15

Redaktion: Prof. Dr. Christoph Stamm
Layout: Claude Rubattel
Erscheinungsweise: jährlich
Druck: jobfactory Basel
Auflage: 150

ISSN 1662-2014 (Print)
ISSN 2296-4169 (Online)

etwa 10 TBytes anfallen. Diese Datenmenge lässt sich auf etwa 2000 DVDs abspeichern. Meine aktuelle persönliche Datensammlung in Büro und zuhause enthält Ende 2017 rund acht Terabytes.

Um die im Jahr 2020 erwartete Datenproduktionsrate von 1.7 MByte/s einordnen zu können, versuche ich hier dieser die maximale Datenrate gegenüberzustellen, die es braucht, damit wir mit all unseren Sinnen gleichzeitig etwas empfinden können. Ich nenne dies die Datenimmissionsrate. Dabei stelle ich mir vor, dass möglichst viele menschliche Sinne in Form einer virtuellen Realität gleichzeitig beeinflusst werden. Dazu braucht es Daten zur Modellierung und Beeinflussung der menschlichen Sinneswahrnehmungen: der visuellen Wahrnehmung mittels stereoskopischer Videos, der auditiven Wahrnehmung mittels binauraler Töne, der haptischen Wahrnehmung (tastendes Begreifen), der Kinetik (Bewegung des Körpers), der Thermorezeption (Wärmeempfindung) der olfaktorischen Wahrnehmung mit der Nase und der gustatorischen Wahrnehmung mit der Zunge. Mit einer Datenrate von ca. 5 MByte/s sollten alle der aufgelisteten Sinneswahrnehmen (mit Ausnahme der beiden letzten, da entsprechende digitale Modelle noch fehlen) modelliert und gesteuert werden können. Der Hauptanteil (ca. 60%) dieser Daten wird dabei für die Beeinflussung der visuellen Wahrnehmung benötigt. Gönnen wir den Menschen täglich noch acht Stunden Schlaf, so können wir für diese Zeit die visuelle Wahrnehmung subtrahieren und kommen auf eine maximale Datenimmissionsrate von ca. 4 MByte/s.

Um also tagsüber fast vollständig immersiv in eine virtuelle Welt einzutauchen, braucht es die durchschnittliche Datenproduktion von 2.4 Personen. Etwas sehr plakativ ausgedrückt: der gleichzeitige Datenausstoss der anderen 4 199 999 997 menschlichen Datenproduzenten wird spurlos an uns vorbeigehen. Das stimmt in dieser Form natürlich nicht ganz, denn es ist sehr viel wahrscheinlicher, dass wir mehrfach aggregierte Daten konsumieren, die von einer viel breiteren Personenbasis stammen. Zudem sind diese Datenraten schwierig miteinander zu vergleichen, weil nichts über den Informationsgehalt der Daten bekannt ist. Trotzdem sollte deutlich werden, dass es nicht viel Sinn ergibt, die Datenproduktionsrate noch weiter zu erhöhen, weil wir gar nicht mehr in der Lage sein werden, diese Daten wahrzunehmen.

Diese Datenimmissionsrate als Pendent zur Produktionsrate verliert natürlich dann ihre Bewandtnis, wenn die produzierten Daten gar nicht mehr für uns Menschen gedacht sind, sondern nur für Maschinen (Computer) produziert werden, also hauptsächlich von Maschine zu Maschine weitergereicht werden.

Um Orientierung in den grossen Datenberg zu bringen, wird mittels Algorithmen zwischen rohen und sinnstiftenden Daten, sogenannten *Smart Data,* unterschieden. Damit soll ein qualitativer Unterschied zwischen langweiligen und ermüdenden Zahlenreihen und verständlichen Daten, deren Sinn sich dem Nutzer sofort erschliesst, gemacht werden. Smarte Daten können sowohl benutzt werden, um unter Nutzung von Rohdaten neue Erkenntnisse zu gewinnen, als auch um neue Modelle zu schaffen, die für die Analyse von Rohdaten genutzt werden können. Seien der Verlauf eines Aktienkurses die Rohdaten, dann sind der Trend, die Extremwerte und spezielle Ausschläge die smarten Daten. Dabei ist wichtig zu sehen, dass Smart Data auf einer Metastufe wiederum Rohdaten darstellen, welche auf einer Meta-Metastufe veredelt werden können usw.

Wer meint, dass auf Halde produzierte Rohdaten gelöscht werden, nachdem sie analysiert und sinnvolle Informationen daraus abgeleitet worden sind, der unterschätzt die Kraft von «man weiss ja nie, wozu man diese Daten noch brauchen kann». «Löschen» von Daten geschieht auf eine viel subtilere Art und Weise. Nämlich dadurch, dass grosse Datenbestände auf Speichermedien gelagert werden, die ausser Mode geraten sind und nur noch mit erheblichem Aufwand auf aktuelle Speichertechniken übertragen werden können. Haben Sie noch irgendwo Floppy-Disks herumliegen? Falls ja, dann verstehen Sie wohl, was ich meine.

Anstatt aus früher gespeicherten Rohdaten zu einem späteren Zeitpunkt smarte Daten zu generieren und damit den Datenberg noch weiteranwachsen zu lassen, versteht man unter *Fast Data*, dass die Rohdaten quasi in Echtzeit unmittelbar nach ihrer Produktion analysiert werden und nur die Erkenntnisse zurückbleiben, während die Rohdaten so schnell wie sie produziert worden sind auch wieder gelöscht werden. Es geht hier also um *Streaming* anstatt *Storing*. Bei Fast Data überwiegt der Nutzen der Echtzeitanalyse gegenüber der Historisierung der Daten. Handelsentscheidungen bei Aktien beispielsweise werden in Echtzeit anhand der Analyse der aktuellen Kursverläufe und Metadaten über die Vergangenheit getroffen, also eine Kombination aus Fast und Smart Data.

An der FHNW in Brugg-Windisch startet im Herbst 2019 der Studiengang *Data Engineering*, der zukünftige Absolventen befähigen wird, den digitalen Wandel aktiv mitzugestalten. Dabei steht thematisch die praxisorientierte *Data Science* im Mittelpunkt. Es ist davon auszugehen, dass *Data Engineering* zukünftig in fast allen Branchen eine zentrale Rolle einnehmen wird. In diesem Sinne entspricht das Profil einem «Ingenieurwesen» im digitalen Zeitalter. Nebst den zukunftsorientierten Inhalten ist auch das Ausbildungskonzept auf die digitale Zukunft ausgerichtet und ermöglicht ein individualisiertes und flexibles Studium.

Prof. Dr. Christoph Stamm

Der Blockchain Hype

Die Kryptowährung Bitcoin ist weltweit bekannt. Die darunterliegenden Konzepte und Technologien sind aber vielen Leuten unbekannt. Der Begriff Blockchain hingegen sagt meist nur technisch interessierten Personen etwas. Die Blockchain ist die Technologie, welche Kryptowährungen wie Bitcoin überhaupt erst ermöglicht. Kryptowährungen und vor allem Blockchains durchlaufen gerade eine Hype-Phase und immer mehr Projekte werden gestartet, die mit Blockchains zu tun haben. In diesem Artikel beschreiben wir, was es mit dem ganzen Hype auf sich hat, wie die tragende Technologie dahinter funktioniert und was sich damit alles bewerkstelligen lässt.

Markus Knecht | markus.knecht@fhnw.ch

Die Idee von *Bitcoin* [1] und der *Blockchain* wurde 2008 von Satoshi Nakamoto in [2] publiziert. Satoshi Nakamoto ist ein Pseudonym und es ist unklar, welche reale Person, Gruppe oder Firma dahintersteckt [3]. Seit der Erfindung von Bitcoin hat sich vieles getan im Blockchain-Bereich: In den letzten Jahren sind sehr viele neue Kryptowährungen aufgekommen, wovon bereits mehr als 800 auf verschiedensten Handelsplätzen gehandelt werden [4]. Die meisten dieser Kryptowährungen bieten diverse Vorteile gegenüber Bitcoin. Eine Auswahl der Bekannteren ist in Abbildung 1 aufgeführt.

Mittlerweile hat auch die Industrie Interesse an Blockchains gefunden, aber weniger bezüglich der Möglichkeit, eigene Währungen zu erstellen, sondern vorrangig wegen der Eigenschaft, dass auf Blockchain basierte Applikationen ohne vertrauenswürdige Instanz auskommen können. Dies hat das enorme Potenzial, ganze Abteilungen oder Firmen aus bestimmten Prozessabläufen zu entfernen, ohne eine Einbusse der Vertrauenswürdigkeit zu erleiden.

Blockchains werden häufig als ultimative Technologie für dezentrale Applikationen vermarktet und Projekte, die auf Blockchains aufbauen, erhalten relativ leicht grosse Projektmittel. Einige Beispiele solcher Projekte sind in Tabelle 1 aufgelistet. Dazu verwenden sie oft eine neue Art von Crowdfunding, dem sogenannten *Initial Coin Offering* (ICO), welches eine auf Kryptowährungen basierte Art des Crowdfundings ist. Ein ICO hat zum Ziel, die Entwicklung eines Produkts zu finanzieren. Bei einem ICO werden Münzen einer neuen digitalen Währung verkauft. Meist kann nur mit anderen Kryptowährungen bezahlt werden, es gibt aber einige Drittfirmen wie zum Beispiel die Bitcoin Suisse AG [10], welche Investitionen in bestimmten Fiat-Währung[1] ermöglichen. Die in einem ICO erstellte neue Währung ist projektspezifisch und kann verwendet werden, um Services im Endprodukt zu beziehen. Einige ICO verfolgen einen klassischeren Ansatz und geben den Besitzern der Währung relativ zu ihrem Kapital Stimmrechte und/oder schütten Dividenden aus. Da die ICO Währung auch gehandelt werden kann, hängt ihr Wert stark von den Erwartungen in das entsprechende Projekt ab und verhält sich in diesem Aspekt ähnlich wie eine Aktie. Die Preisschwankungen sind jedoch oft sehr viel grösser als es bei Aktien der Fall ist.

Das momentane Investitionsverhalten für Blockchain-Projekte wird häufig mit dem Investitionsverhalten bei Internetprojekten während der Dotcom-Blase verglichen. Es wird in diesem Zusammenhang häufig vom Internet of Value [19] gesprochen. Die Erwartung ist, dass Blockchains einen Paradigmenwechsel einleiten wird in Bezug auf die Transferierung von Besitztümern, ähnlich dem Paradigmenwechsel, der durch das Internet in Bezug auf die Übertragung von Informationen stattgefunden hat. Bei der Frage ob sich der Kryptowährungsmarkt in einer Blase befindet oder nicht, gehen die Meinungen auseinander, da das

Abbildung 1: Auswahl grösserer Kryptowährungen gemessen am Marktwert: Bitcoin [1], Ethereum [5], Litecoin [6], Ripple [7], Dash [8], Monero [9]

1 Währung ohne intrinsischen Wert, welche von einer Regierung herausgegeben wird

Projekt	Land	Volumen	Dauer
Tezos [11]	Schweiz	206	14 Tage
Bancor [12]	Schweiz	153	3 Stunden
Status [13]	Schweiz	95	17.2 Stunden
MobileGo [14]	USA	53	1 Monat
Basic Attention Token [15]	USA	35	30 Sekunden
Polybius [16]	Estland	31	6 Wochen
Aragon [17]	Estland	25	15 Minuten
Aeternity [18]	Lichtenstein	24	3 Wochen

Tabelle 1: Beispielhafte Auflistung abgeschlossener ICOs (Volumen in Millionen Dollars)

ökonomische Verhalten[2] von Kryptowährungen so unterschiedlich ist zu klassischen Märkten, dass eine Analyse schwerfällt.

Können Kryptowährungen bereits verwendet werden?

Die rapide Entwicklung von Kryptowährungen hat auch zur Folge, dass es ständig mehr Möglichkeiten gibt, um Kryptowährungen zu erwerben und damit Einkäufe zu machen. In Zug befindet sich das sogenannte *Crypto Valley* [20], begrifflich angelehnt am Silicon Valley, wodurch die Schweiz eine führende Rolle in der Entwicklung von Blockchain-Technologien einnimmt und viele namhafte Blockchain-Firmen und Stiftungen dort ihren Sitz haben. Dies hat die Gemeinde Zug motiviert, Bitcoins als Zahlungsmittel für Gemeindedienstleistungen bis zu 200 CHF anzunehmen [21]. In der Schweiz ist es sehr einfach Bitcoins zu erwerben, da jeder SBB-Automat diese zum Verkauf anbietet [22]. Auch sogenannte Bitcoin-Geldautomaten gibt es bereits einige über die ganze Schweiz verteilt. Die erworbenen Bitcoins können dann schnell, bequem und günstig über einen Krypto-Exchange, wie zum Beispiel den Schweizer Anbieter *Shapeshift* [23], in andere Kryptowährungen getauscht werden. Möchte man seine Kryptowährungen einfach und bequem im Alltag verwenden können so kann man sich eine Krypto Kredit/Debit Karte anschaffen und dann überall bezahlen, wo mit herkömmlichen Karten wie *Visa* und *Master Card* bezahlt werden kann. Zu guter Letzt akzeptieren immer mehr Anbieter von Produkten und Services Kryptowährungen als Zahlungsmittel.

2 Da es in Kryptowährungen im Gegensatz zu Fiat-Währungen keine zentrale Autorität (Zentralbank) gibt, welche neues Geld erschafft und Zinssätze festlegt, haben Kryptowährungen ein anderes, noch unerforschtes ökonomisches verhalten. Zudem existieren noch praktisch keine Erfahrungswerte aus der Vergangenheit, an denen man sich orientieren könnte. Für den interessierten Leser bieten folgende Artikel Informationen aus ökonomischer Sicht zu diesem Thema:
https://www.cryptocoinsnews.com/digital-currencies-bringing-reality-hayeks-free-market-money/
https://benbest.liberty.me/persuading-austrian-economists/

Wieso dieser Hype, was bietet die Blockchain?

Bei den Blockchains kann man zwischen öffentlichen und privaten Blockchains unterscheiden. Private Blockchains sind im Gegensatz zu öffentlichen nicht für jedermann zugänglich und werden entwickelt, um einen Dienst für einen geschlossenen Benutzerkreis bereitzustellen. Vergleicht man Blockchains mit der Internettechnologie, so sind die öffentlichen Blockchains mit dem Internet vergleichbar und die privaten mit einem Intranet. Viele Puristen betrachten private Blockchains oft nicht als reale Blockchains, da diese teilweise stark von der ursprünglichen Idee abweichen und in vielen Bereichen Kompromisse eingehen. Im Folgenden werden öffentliche Blockchains beschrieben, da diese besser vereinheitlicht werden können.

Eine Blockchain ist ein Protokoll, das einen Zustand verwaltet, welcher sich nur nach festgelegten Regeln verändern lässt. Eine Veränderung wird über deterministische Transaktionen von einem Teilnehmer ausgelöst. Dieser Zustand ist dezentral auf den Geräten der Nutzer der Blockchain abgelegt, wobei sich alle über den aktuellen Zustand einig sind und diesen nur anhand von vorgegebenen Regeln verändern können. Eine Blockchain hat diverse Eigenschaften und bietet Garantien, die Anwendungen ermöglichen, welche ohne diese Garantien nicht realisiert werden könnten.

In einer öffentliche Blockchain kann jeder mitmachen (*open*), egal wo er ist (*borderless*) und wie er mit der Blockchain interagiert (*censorship resistant*). Man muss keine persönlichen Daten preisgeben um teilzunehmen (*pseudonym*). Es gibt keine privilegierten Teilnehmer: für jeden gelten die gleichen Regeln, ob Mensch oder Kühlschrank (*neutral*). Die Regeln können nicht gebrochen werden (*uncorruptable*). Es muss keiner zentralen Instanz vertraut werden, um sich sicher zu sein, dass das System korrekt funktioniert (*trustless*). Die Blockchain kann nicht abgestellt werden, solange es Teilnehmer gibt, die sie weiterführen wollen (*unstoppable*). Eine bestätigte regelkonforme Änderung kann nicht mehr rückgängig gemacht werden (*immutable*) und ist für jeden für immer einsehbar (*transparent*).

Wie funktioniert eine Blockchain?

Damit alle Teilnehmer den selben Zustand verwenden ist es essentiell, dass jeder dieselben Transaktionen in derselben Reihenfolge verarbeitet. Dazu werden die Transaktionen in Blöcke zusammengefasst, wobei die Transaktionen in jedem Block eine klare Reihenfolge aufweisen. Neben den Transaktionen enthält jeder Block einen Verweis auf einen anderen bereits existierenden Block und bildet somit eine verkettete Liste (Abb. 2). Alle Transaktionen im Block auf den verwiesen wird, sind vor den Transaktionen im verweisen-

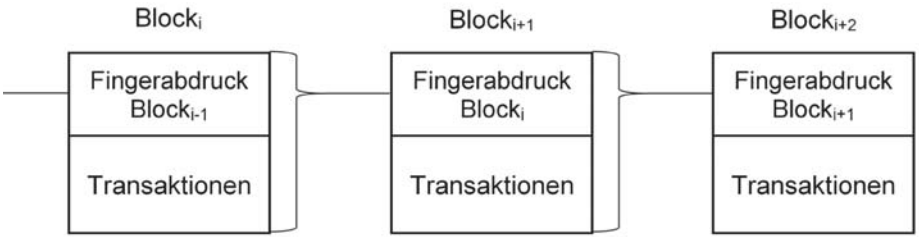


Abbildung 2: Grundlegende Struktur der Blockchain

den Block angeordnet. Der Verweis wird durch einen Fingerabdruck (*Hash*-Wert) des Inhaltes des Vorgängerblocks dargestellt, was mit ausreichend hoher Wahrscheinlichkeit sicherstellt, dass der Inhalt des Vorgängers nicht mehr geändert werden kann, ohne auch den Nachfolger zu ändern.

Falls nun das Erstellen eines Blockes mit einem grossen Ressourcen-Aufwand verbunden wird, so wird die Fälschung eines existierenden Blockes umso aufwendiger, je weiter entfernt er vom neusten Block in der Blockkette entfernt ist. Dieser Ressourcen-Aufwand wird in Bitcoin mit dem sogenannten *Proof of Work* (einem Consensus Algorithmus) erreicht. Andere Blockchains verwenden Alternativen wie zum Beispiel *Proof of Stake*.

Im Proof of Work versuchen sogenannte *Miner* ein rechenintensives Rätsel (ein sogenanntes Hash-Puzzle) zu lösen, welches abhängig ist vom Block, den er produzieren möchte. Die effizienteste Variante das Rätsel zu lösen erfolgt durch die Erzeugung von Zufallszahlen und der Überprüfung, ob diese eine Lösung darstellen. In Bitcoin wird die Schwierigkeit des Rätsels automatisch angepasst, so dass im Durchschnitt alle zehn Minuten jemand eine Lösung findet, sprich einen neuen Block produzieren kann. Für das Lösen des Rätsels und das Generieren eines Blocks gibt es eine Belohnung in Bitcoins, sofern der Block als Teil der Hauptkette angesehen wird. Zudem zahlt der Ersteller einer Transaktion eine Gebühr an denjenigen, der sie in einen Block integriert. Als Hauptkette wird diejenige Kette mit den meisten Blöcken angesehen, welche nur Transaktionen enthält, die eine gültige Zustandsveränderung herbeiführen und jede Transaktion nur einmal beinhaltet. Die Miner sind somit dem Anreiz ausgesetzt, die momentane Hauptkette zu erweitern und nur Blöcke mit gültigen Transaktionen und ohne Duplikate zu generieren, da sie sonst ohne Belohnung ausgehen.

Ein Block, der Teil der Hauptkette ist, kann nur dann aus der Hauptkette wegfallen und damit die darin enthaltenen Transaktionen nichtig machen, wenn jemand eine neue Kette erstellt, die vor diesem Block abzweigt und länger ist als die momentane Hauptkette. Dies erfordert jedoch, dass er mehr Rechenleistung zur Rätsellösung einsetzt als all diejenigen, die die Hauptkette erweitern, damit seine Kette irgendwann länger als die momentane Hauptkette wird. Da das Rätsellösen einen Glücksfaktor beinhaltet, können kurze Überholmanöver auch mit weniger Rechenleistung vorkommen, was zur Folge hat, dass neue Blöcke erst nach einer gewissen Zeit als final angesehen werden. In Bitcoin wartet man, bis fünf weitere Blöcke auf einem Block aufbauen, bis man diesen als final und somit quasi-unveränderbar ansieht.

Was kann mit einer Blockchain gemacht werden?

Im folgenden Abschnitt realisieren wir anhand der gegebenen Definitionen eine beispielhafte Kryptowährung. Die entsprechende Blockchain vermerkt als Zustand, welcher Teilnehmer wie viele der Kryptomünzen besitzt. Die Regeln um diesen Zustand zu verändern besagen, dass wenn sich jemand authentifizieren kann (mit Hilfe von Kryptographie), dass er seine Anzahl Münzen verringern kann und die eines anderen Teilnehmers um dieselbe Anzahl erhöhen kann, solange seine Anzahl dabei nicht negativ wird. Solche Regeln laufen immer als Ganzes (atomar) ab und im vorherigen Beispiel ist deshalb garantiert, dass Münzen nur transferiert, jedoch nicht dupliziert werden können, wodurch ihre Anzahl ständig konstant bleibt.

Neben Kryptowährungen gibt es viele andere Anwendungen die mithilfe von Blockchains realisiert werden können:

- *Lunyr*: Dezentrale Wissensdatenbank, ähnlich zu Wikipedia aber mit einem finanziellen Anreizsystem, um die Qualität der Artikel mithilfe von *Peer Reviews* ständig zu verbessern [28].
- *BitGive*: Eine dezentrale Hilfsorganisation, die das Spenden erleichtert und es erlaubt transparent nachzuverfolgen, dass die Spende auch wirklich für den angedachten Zweck eingesetzt wird [29].
- *uPort*: Digitale Identitätsplattform, welche die Kontrolle der Identität zurück an das Individuum gibt [30]. Die Gemeinde Zug arbeitet mit *uPort* zusammen, um für seine Bürger eine digitale Identität zur Verfügung zu stellen [31].
- *Arcade City*: Dezentralisierte Plattform zum Etablieren einer *Sharing Economy*, beginnend mit einem dezentralisierten Service vergleichbar mit Uber [32].
- *Slock.it*: Smart Lock Projekt um Schlösser sowie Öffnungsberechtigungen dynamisch auf der Blockchain zu verwalten, um Eigentumstransfers und Ausleihen zu digitalisieren [25].

- *Everledger*: Dezentralisierte Verfolgung von Diamanten, um Betrug und andere Risiken zu reduzieren «Schlüsselwort Blutdiamanten» [33].

Smart Contracts

Einige Blockchains, wie zum Beispiel *Ethereum* [5], machen ihre Blockchain durch sogenannte *Smart Contracts* programmierbar, was erlaubt, Subsysteme einzuführen, die ihren eigenen Zustand und Regeln haben. Die Struktur des Zustands und die Regeln, um diesen zu verändern, werden mithilfe einer Programmiersprache beschrieben. Listing 1 zeigt, wie eine neue Kryptowährung mit minimalem Funktionsumfang als Smart Contract auf Ethereum als Subsystem implementiert werden kann. Ethereum verwendet als Programmiersprache *Solidity* [24]. Jeder kann eine Instanz dieses Smart Contracts erstellen, um seine eigene Kryptowährung zu generieren, wobei dann jede Instanz eine unterschiedliche und eigenständige Währung repräsentieren würde.

Blockchains sind jedoch nicht auf Kryptowährungen beschränkt, sondern können auch dazu verwendet werden, um alle möglichen Arten von Besitz zu verfolgen, wie zum Beispiel ein Haus. Hat das Haus sogar noch ein *Smart Lock* [25], das mit der Blockchain verbunden ist, könnte es seinen momentanen Besitzer kennen und nur diesem (und von ihm autorisierte Personen) Zutritt zum Haus gewähren. Sobald der Besitz digital repräsentiert werden kann, eröffnen sich neue Möglichkeiten, wie zum Beispiel eine dezentrale *AirBnB*-Variante [26]. Um dies zu bewerkstelligen, würde ein Smart Contract eine Zutrittserlaubnis für einen bestimmten Zeitraum ausstellen, falls ein entsprechender Auftrag existiert und die dazugehörende Zahlung eingegangen ist. Das Smart Lock kann bei einer Öffnungsanfrage nun überprüfen, ob die entsprechenden Person die Erlaubnis besitzt um einzutreten. Das Interessante an einer solchen Lösung ist, dass eine Institution wie AirBnB als zentrale und verwaltende Instanz überflüssig gemacht werden kann, was das System robuster gegen Angriffe und Einflüssen von Drittparteien macht. Zudem kann die Kosteneffizienz verbessert werden.

Durch das geschickte Zusammenspiel von IoT-Geräten mit einer Blockchain eröffnen sich neue, futuristische und bisher undenkbare Anwendungen: Ein mit einer Blockchain verbundenes, selbstfahrendes, elektrisches Auto könnte einen Service ähnlich zu *Uber* [27] anbieten. Das autonome Auto nimmt über das Internet Aufträge entgegen und holt dann selbständig die Kunden ab, welche während der Fahrt kontinuierlich für jede Minute und/oder Kilometer über eine Kryptowährung direkt das Auto bezahlen. Mit den Einnahmen bezahlt das selbstfahrende Auto seine Auslagen, wie zum Beispiel Reinigungen, Reparaturen, Serviceüberprüfung und das Aufladen der Batterie. Nebenbei zahlt es von den Einnahmen noch die Schulden für ein Darlehen ab, mit dem seine Produktion finanziert worden ist. Damit ein solches Szenario seitens der Blockchain möglich wird, müssen zuerst einige Veränderungen in Bezug auf die Akzeptanz von Kryptowährungen zur Bezahlung von diversen Ressourcen, Gütern und Dienstleistungen (z.B. Reparaturen und Reinigungen) erfolgen.

Eignung für alle dezentralen Applikationen?

Blockchains habe auch diverse Nachteile, zum einen haben sie momentan noch einen geringen Transaktionsdurchsatz. Bitcoin kann zum Beispiel nur etwa sieben Transaktionen pro Sekunde verarbeiten, weit entfernt von den mehreren tausend die Visa aktuell pro Sekunde abwickeln kann. In diesem Bereich werden jedoch ständig Fortschritte gemacht und neuere Blockchains sind

```
// Example Smart contract
contract FokusReportToken {
  // adds overflow save arithmetic operations to unsigned integers
  using SafeMath for uint;

  // balances, tracks who (address) owns how many (uint) coins
  mapping(address => uint) balances;

  // constructor
  function FokusReportToken() {
    // creator of contract (msg.sender) gets all the initial tokens
    balances[msg.sender] = 10*1000*1000;
  }

  // transfers some coins (value) from authenticated sender (msg.sender) to somebody else (to)
  function transfer(address to, uint value) {
    balances[msg.sender] = balances[msg.sender].sub(value);
    balances[to] = balances[to].add(value);
  }
}
```

Listing 1: minimale funktionsfähige Kryptowährung in Solidity

performanter. Viele Projekte versprechen momentan eine Performance, die mit der Transaktionsmenge der Grössenordnung von Visa oder sogar noch grösser klarkommen könnte. Diese Projekte sind aber erst in der frühen Entwicklungsphase oder sehr spezialisiert, wie zum Beispiel nur für Bezahlvorgänge aber nicht für Smart Contracts verwendbar.

Ein weiterer Nachteil ist, dass der Speicherbedarf der Blockchain ständig wächst und droht, mit der Zeit zu gross zu werden für herkömmliche Endnutzergeräte. Die Bitcoin Blockchain braucht momentan etwas mehr als 130 GByte an Speicherplatz und diese Zahl wächst linear mit der Zeit. Solange *Moore's Law* anhält, welches voraussagt, dass sich die Speicherkapazität von Speichermedien alle 18 Monate verdoppelt bei gleichem Preis, ist die linear wachsende Blockchain für Nicht-Endnutzergeräte unproblematisch. Zudem werden auch in diesem Bereich der Blockchain ständig Fortschritte gemacht: *Light Clients* und *Sharding*-Ansätze [34] ermöglichen es, dass ein Teilnehmer nur einen kleinen Teil der Blockchain speichern muss, ohne Sicherheitseinbussen in Kauf nehmen zu müssen.

Viele der zuvor erwähnten Eigenschaften (*open*, *borderless*, etc.) sind leider zusammenhängend und können nicht individuell ab oder angeschaltet werden. Möchte man zum Beispiel auf *transparent* verzichten und stattdessen mehr *privacy* haben, weil ein Pseudonym nicht ausreichend ist, so ist das nicht ohne weiteres möglich. Hat man die Anforderung, dass man auf Anfrage bestimmte Inhalte entfernen können muss, dann werden *censorship resistant* und *immutable* sowie *pseudonym* plötzlich zu einem Hindernis. Braucht eine Applikation nicht alle Eigenschaften, sollte man sich gut überlegen, ob die Blockchain die richtige Lösung ist oder ob man die Applikation nicht doch besser mit einer anderen Technologie realisiert. Sobald ein Teilnehmer einem Dienstleister vollständig vertrauen muss, wird die Blockchain in den meisten Fällen unnötig, da der Vertrauenspartner all die anderen benötigten Eigenschaften mithilfe einer auf einem vertrauenswürdigen Server laufenden Applikation sicherstellen kann. Ihr volles Potenzial entfaltet eine Blockchain-Applikation somit erst, wenn sie eine Dienstleistung zur Verfügung stellt, die ohne zusätzlich involvierte juristische Person funktionsfähig sein soll. Bitcoin und andere Kryptowährungen sind Paradebeispiele, da sie die Dienstleistung des Geldtransfers anbieten, ohne dass eine Bank oder ein anderes Institut involviert werden muss.

Private Blockchains versuchen momentan Lösungen zu finden, die Kompromisse eingehen, um bestimmte Eigenschaften wie *privacy* und *governance*, sowie *permissioning* zur Verfügung zu stellen, so dass Blockchains auch in Firmen eingesetzt werden können, wo Regulatoren verlangen, dass die Applikation bestimmte Eigenschaften hat, welche eine öffentliche Blockchain nicht zur Verfügung stellen kann.

Referenzen

[1] Bitcoin: http://www.bitcoin.org

[2] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. http://www.bitcoin.org/bitcoin.pdf, 2008.

[3] Wer ist Satoshi Nakamoto: Die moderne Jagd nach Mr.X: https://www.krypto-magazin.de/wer-ist-satoshi-nakamoto-die-moderne-jagd-nach-mr-x/

[4] Coinmarketcap: https://coinmarketcap.com/

[5] Ethereum: https://www.ethereum.org/

[6] Litecoin: https://litecoin.com/

[7] Ripple: https://ripple.com/

[8] Dash: https://www.dash.org/

[9] Monero: https://getmonero.org/

[10] Bitcoinsuisse: https://www.bitcoinsuisse.ch/

[11] Tezos: https://www.tezos.com/

[12] Bancor: https://www.bancor.network/

[13] Status: https://status.im/

[14] Mobile Go: https://mobilego.io/

[15] BAT: https://basicattentiontoken.org/

[16] Polybius: https://polybius.io/

[17] Aragon: https://aragon.one/

[18] Aethernity: https://www.aeternity.com/

[19] Internet of Value: https://ripple.com/insights/the-internet-of-value-what-it-means-and-how-it-benefits-everyone/

[20] Crypto Valley: https://cryptovalley.swiss

[21] Von Bitcoin zu Blockchain-Anwendungen: http://www.stadtzug.ch/de/ueberzug/ueberzugrubrik/aktuelles/aktuellesinformationen/welcome.php?action=showinfo&info_id=351680

[22] Mit Bitcoin bequem und einfach Einkaufen: https://www.sbb.ch/de/bahnhof-services/dienstleistungen/weitere-dienstleistungen/bitcoin.html

[23] Shapeshift: https://shapeshift.io

[24] Solidity: http://solidity.readthedocs.io/en/develop/index.html

[25] Slock.it: https://slock.it

[26] AirBnB: https://www.airbnb.ch

[27] Uber: https://www.uber.com

[28] Lunyr: https://lunyr.com

[29] Bitgive: https://www.bitgivefoundation.org/

[30] Uport: https://www.uport.me

[31] Blockchain-Identität für alle Einwohner: http://www.stadtzug.ch/de/ueberzug/ueberzugrubrik/aktuelles/aktuellesinformationen/?action=showinfo&info_id=383355

[32] Arcade City: https://arcade.city

[33] Everledger: https://www.everledger.io

[34] What is Sharding? https://themerkle.com/what-is-sharding/

Scoop – Mobile Payment

In diesem Artikel wird die elektronische Geldbörse Scoop vorgestellt, die im Rahmen eines KTI Projektes entwickelt worden ist. Im Gegensatz zu anderen Mobile Payment Lösungen muss bei Scoop der Point of Sale (POS) nicht mehr direkt an das Internet angeschlossen sein. Das Settlement der Zahlung wird direkt am POS vorgenommen und, falls das Mobiltelefon am POS zum Zeitpunkt der Zahlung nicht online ist, so werden die Transaktionsdaten über ein langsames Netzwerk an den Server ausgeliefert. Um «Double-Spend»-Attacken zu erkennen, werden alle Transaktionen in einer Kette (Chain) gespeichert und mit kryptographischen Hash-Codes gesichert. Im POS sind die Hash-Codes in einem mit JavaCard programmierten Secure-Element gespeichert. In diesem Artikel präsentieren wir die Idee von Scoop und erläutern Aspekte der Realisierung.

Dominik Gruntz, Markus Knecht, Stephan Wullschleger[1] | dominik.gruntz@fhnw.ch

Im Rahmen des Projektes *Supervised Chaining Offline/Online Purse* (Scoop) haben wir zusammen mit der Firma PBV Kaufmann Systeme GmbH eine Payment-Lösung entwickelt, mit der sichere Offline-Transaktionen abgewickelt werden können. Man kann sich zu Recht fragen, ob es neben *Apple Pay*, *Google Pay*, *Samsung Pay*, *Huawei Pay*, *Alipay*, *Twint*, *Boon*, *Seqr Go!*, *Postfinance*, *Visa Bonus Card App*, *MIGROS App* etc. wirklich noch eine weitere *Mobile Payment* Lösung braucht. Die existierenden Lösungen haben jedoch alle eines gemein: Der *Point of Sale* (POS) muss online mit dem Internet verbunden sein, damit bei der Zahlung mit dem Mobiltelefon das Zahlungsterminal die Kartengültigkeit, die Karten- und Tageslimite sowie eine ausreichende Kontodeckung überprüfen kann. Die Zahlung muss also vom Server freigegeben werden [Mau09].

Es macht jedoch nicht für jeden POS Sinn eine Internet-Connectivity sicherzustellen, auch wenn diese mit einem einfachen GSM-Modul realisiert werden könnte. Wir denken da insbesondere an Kaffeeautomaten oder Snackautomaten wie sie häufig auch in Firmen anzutreffen sind oder auch an Waschautomaten. Für solche Anwendungen wurden kartenbasierte elektronische Geldbörsen wie z.B. die *GeldKarte* in Deutschland, *CASH* in der Schweiz oder *Quick* in Österreich entwickelt. Sowohl CASH wie auch Quick sind inzwischen jedoch eingestellt worden. Die Einstellung von Quick per Mitte 2017 hat jedoch zu Problemen in Waschküchen sowie bei vielen Zigaretten-, Snack- und Parkautomaten geführt [Stau17]. Die Umstellung auf eine Online-Lösung ist gerade in Waschküchen wegen der fehlenden Internet-Verbindung nicht immer möglich.

Mit Scoop haben wir uns auf den Anwendungsbereich der bargeldlosen Offline-Zahlungen fokussiert. Offline-Zahlung heisst bei uns, dass sowohl das Payment-Terminal (also der eigentliche POS) wie auch das Mobiltelefon zum Zeitpunkt der Zahlung offline sein können.

Offline-Payment Systeme

Bei einer Geldkarte ist das Guthaben auf der Karte gespeichert, und wenn an einem POS mit der Karte bezahlt wird, dann wird der entsprechende Betrag vom Kartenguthaben abgezogen und auf dem POS gutgeschrieben. Falls das bezahlte Produkt nicht ausgegeben werden kann, so wird der Betrag auf die Karte zurückgebucht (falls die Karte noch zugreifbar ist) oder es wird eine Storno-Transaktion auf dem POS gespeichert, die vom Karteninhaber am POS abgeholt werden kann. Falls diese Transaktion jedoch nicht abgeholt und gelöscht wird, dann ist das Geld verloren. Geld geht auch dann verloren, falls ein Nutzer seine Karte verliert oder falls ein Betrag von der Karte abgezogen wird, der POS die Bestätigung dieser Transaktion aber nicht mehr erhält. Man spricht in diesem Zusammenhang auch von einem Schlupf, da so potentiell Geld aus dem System entweichen kann.

Online-Payment Systeme

Das geschilderte Schlupfproblem kann gelöst werden, indem die Guthaben auf einem Server gespeichert und die Transaktionen auf diesem Server nachgeführt werden, d.h. bei jedem Bezahlvorgang meldet der Sender oder der Empfänger den Geldübertrag an den Server, und dieser führt diese Transaktion in seiner Registratur nach. Solange der Server nicht informiert ist, ist das Geld nicht verschoben. Guthaben kann so nicht verschwinden und ist immer eindeutig einem Besitzer zugeordnet. Auf der Karte selber wird bei dieser Lösung kein Guthaben gespeichert, sondern nur die Identifikation des Nutzers, und der Bezahlvorgang wird nicht lokal am POS sondern auf dem Server ausgeführt. Die Karte für Zutritt und Bezahlung an der FHNW (FH-Card) verwendet beispielsweise diese Lösung und daher sind

1 PBV Kaufmann Systeme GmbH

die Kassenterminals, die Parking-Schranken und alle Kaffeeautomaten an der FHNW online mit dem Internet verbunden.

Anstelle einer Nutzeridentifikation könnte auf dem Mobiltelefon auch eine Bezahlungsberechtigung gespeichert werden (vergleichbar mit einem Verrechnungsscheck), die dann offline auf ein anderes Gerät oder an einen POS übertragen wird. Die Transaktion erfolgt jedoch erst, wenn der Sender oder der Empfänger die Verschiebung der Berechtigung an den Server meldet. Ein Schlupf wird so vermieden, denn auch wenn ein Check mehrfach verwendet wird, so kann er nur durch einen Empfänger eingelöst werden. Diese Variante ist 2015 von Visa patentiert worden [Sabba16].

Ein ebenfalls auf Bezahlungsberechtigungen basierendes Protokoll wurde bereits 2014 an unserem Institut im Rahmen des Projektes *iBeam* entwickelt [iBeam14]. Die Bezahlungsberechtigungen wurden damals mit einem Verfalldatum versehen. Wenn der Server bis zum Ablauf der Gültigkeit des Checks weder vom Sender noch vom Empfänger über eine Transaktion informiert wurde, so wurde der zurückgestellte Geldbetrag wieder freigegeben.

Scoop

Scoop vereinigt die Vorteile von Offline- und Online-Payment Systemen. Das Guthaben wird auf dem Mobiltelefon (bzw. auf der Karte) und auf dem POS gespeichert. Das Settlement (Austausch von Leistung gegen Geld) wird unmittelbar am POS (offline) ausgeführt und Guthaben kann nicht verloren gehen (d.h. kein Schlupf). Dies wird erreicht, indem Transaktionen mit idempotenten Operationen[2] zwischen verschiedenen Transaktionsketten verschoben werden. Zudem werden die am Scoop teilnehmenden Mobiltelefone genutzt, um Daten vom POS an den Server zu übertragen. Transaktionsdaten werden dabei über alle Mobiltelefone verschickt, die mit dem POS interagieren bis am POS eine Bestätigung eingetroffen ist (Schwarm-Netzwerk).

Auf dem POS wird das Guthaben in einem *Secure Element* (SE) gespeichert (wir verwenden die MicroSD-Karte PS-100u VE mit einem SE von Swissbit). Ein SE ist ein sicherer Applikations- und Datenspeicher (*Trusted Execution Environment*), vergleichbar mit dem Chip auf einer Kreditkarte. Leider stehen entsprechende Elemente auf dem Mobiltelefon nicht zur Verfügung (bzw. können wie beim iPhone nur vom Hersteller genutzt werden). Daher könnte ein gewiefter Nutzer das auf dem Mobiltelefon gespeicherte Guthaben manipulieren. Das Protokoll ist jedoch so ausgelegt, dass solche Betrugsfälle entweder direkt am POS erkannt und zurückgewiesen oder spätestens auf dem Server erkannt und korrigiert werden können. Da wir neben dem Offline-Protokoll auch ein Online-Protokoll unterstützen, bei dem sichergestellt ist, dass das Guthaben nicht überzogen werden kann, wird empfohlen, dass die Offline-Variante nur vertrauenswürdigen Nutzern zur Verfügung gestellt wird (z.B. Nutzern, bei welchen man Zugriff auf eine Kreditkarte hat). Das Scoop-System erlaubt es auch, pro Benutzer eine Offline-Limite zu setzen und der POS kann zudem entscheiden, ob er überhaupt Offline-Transaktionen zulässt.

Die Scoop-Lösung ist als europäisches Patent angemeldet worden [AGKW16]. Im Folgenden werden die Kernelemente des Scoop-Protokolls dargestellt.

Protokoll

Das Scoop-Protokoll basiert auf Transaktionsketten, deren Elemente mit Hilfe von kryptographischen Hash-Funktionen untereinander verlinkt werden. Diese Ketten sind durch die Blockchain-Technologie von Bitcoin und anderen Kryptowährungen inspiriert, aber im Unterschied zu diesen Technologien wird bei Scoop kein dezentraler Konsens benötigt, da ein Server als zentrale Instanz diese Funktion übernehmen kann. Zudem verwenden wir in unserem Protokoll nicht nur eine Kette, sondern mehrere Ketten, die jedoch voneinander abhängen. Diese Ketten ermöglichen es, die Sicherheit im Offline-Fall effizient zu erhöhen, da am POS bestimmte Eigenschaften lokal geprüft werden können.

Jedes Element einer Transaktionskette enthält einen Hash-Wert, der aus den Transaktionsdaten und dem Hash-Wert des Vorgängerknotens mit Hilfe einer kryptografischen Hashfunktion berechnet wird. In Abbildung 1 ist ein einfaches Beispiel einer solchen Transaktionskette dargestellt.

In den Elementen wird neben den Daten ein Index abgelegt. Dies erlaubt es dem Server eine Kette einfach zu rekonstruieren, falls die einzelnen Elemente in einer beliebigen Reihenfolge eintreffen.

Der Vorteil dieser Transaktionsketten ist, dass die Integrität der Struktur sichergestellt ist. Hat man einen Hashwert auf das Ende der Kette (in Abb. 1 z.B. H_4) und jemand ändert die Daten eines Elementes oder einen Hashwert in der Kette, so kann dies festgestellt werden, indem die Hashwerte neu berechnet werden. Entweder ist einer der Hashwerte innerhalb der geänderten Kette falsch, oder man erhält einen anderen Wert auf das Ende der Kette[3].

Bei diesen Transaktionsketten ist es möglich, dass ein Element in mehreren Ketten enthalten

2 Idempotente Operationen führen mehrfach ausgeführt zum selben Ergebnis, wie wenn sie nur einmal ausgeführt werden. Dies garantiert Fehlertoleranz bei Verbindungsunterbrüchen.

3 Mit den aktuell verwendeten Algorithmen könnten mit einer vernachlässigbaren Wahrscheinlichkeit von $1/2^{256} = 8.6 \times 10^{-78}$ zwei unterschiedliche Ketten als gleich erkannt werden.

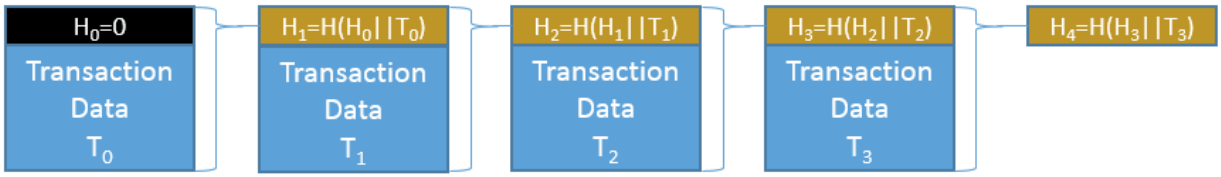


Abbildung 1: Einfache Transaktionskette. H ist eine kryptographische Hashfunktion und || die Konkatenation. T_x repräsentieren die Nutzdaten, H_y die Hashwerte. H_4 ist der Hash-Wert der gesamten Kette.

sein kann, was im Scoop-Protokoll oft verwendet wird. In Abbildung 2 ist ein Beispiel von zwei Ketten dargestellt. Die eine Kette (mit Hashwert A_4) enthält die Transaktionen T_0, T_1, T_2 und T_3, und die zweite Kette (mit Hashwert B_2) enthält nur die Transaktionen T_1 und T_3.

Eine validierende Autorität kann mit einer digitalen Signatur des Hashwerts einer Kette bestätigen, dass die gesamte Transaktionskette existiert und nur gültige Transaktionen enthält. Dies erlaubt es, grosse Transaktionsketten effizient auszutauschen (Hash + Signatur), falls der Empfänger nicht am Inhalt der Kette, sondern nur an deren Existenz und Validität interessiert ist.

Kennt jemand nur den Hashwert einer Transaktionskette, jedoch nicht deren Elemente, so kann er trotzdem prüfen, ob eine zweite Kette mit dieser identisch ist und er kann Elemente an diese Kette anfügen.

Nachfolgend werden die im Scoop-Protokoll verwendeten Transaktionsketten vorgestellt:

- *User-Chain:* Für jedes Mobiltelefon eines Nutzers existiert eine User-Chain, welche das Guthaben des Nutzers auf diesem Mobiletelefon repräsentiert und alle Transaktionen enthält, die das Guthaben erhöhen (Aufladung) oder reduzieren (Übertragung). Diese Kette wird vom Server überwacht und sowohl auf dem Server wie auch auf dem Mobiltelefon gespeichert.
- *POS-User-Chain:* Pro Nutzer und POS existiert eine POS-User-Chain. Diese Kette repräsentiert das von einem Nutzer an einem POS erhaltene (Übertragung) und ausgegebene (Settlement) Guthaben und wird nur auf dem POS gespeichert.
- *Transfer-Chain:* Die Transfer-Chain verbindet die POS-User-Chain mit der zugehörigen User-Chain und repräsentiert alle Transaktionen von Guthaben aus der User-Chain in die POS-User-Chain (Übertragung). Auf dem Mobiltelefon wird ein vom POS angeforderter Betrag von der User-Chain in die Transfer-Chain verschoben und damit für einen spezifischen POS freigegeben. Diese Kette wird danach mit jener auf dem POS synchronisiert, welcher die Einträge nutzt, um die POS-User-Chain zu aktualisieren.

Will ein Nutzer eine Bezahlung durchführen, so fügt er eine Übertragung in die Transfer- und User-Chain ein und synchronisiert dann die Transfer-Chain mit dem POS, wodurch die Übertragung in der POS-User-Chain landet. Schlägt dieser Schritt fehl (verlorene Nachricht, Verbindungsunterbruch etc.), so kann er wiederholt werden, bis die Ketten auf beiden Seiten identisch sind. Über ein Settlement kann nun der POS die Übertragung verwenden, um eine Leistung zu bezahlen. Falls jedoch eine gewünschte Leistung am POS nicht erbracht werden kann, so kann der Nutzer die Übertragung an diesem POS anderweitig verwenden. Nach einer gewissen Zeit wird dieses Guthaben jedoch zurückgesetzt und via Schwarm-Netzwerk und Server wieder in die User-Chain eingefügt.

Sicherheit

Falls auf dem Mobiltelefon eine Manipulation vorgenommen worden ist, um ein Guthaben z.B. mehrfach an einem POS ausgeben zu können (Double-Spend-Angriff), so wird dieser Angriff am POS erkannt, da die Transfer-Chain, die auf dem POS gespeichert ist, Elemente enthält, die auf dem Mobiltelefon nicht vorhanden sind. Es genügt dabei, den auf dem POS gespeicherten Hash-Wert des letzten Elements der Transfer-Chain mit dem im neuen, vom Mobiltelefon gelieferten Kettenelement abgelegten Hash-Wert zu vergleichen.

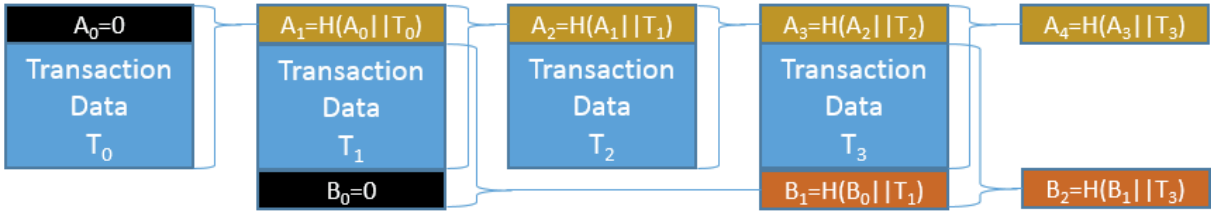


Abbildung 2: Zwei Transaktionsketten, welche die Elemente T_1 und T_3 teilen. T_x repräsentieren die Nutzdaten, A_y die Hashwerte von Kette A, B_y die Hashwerte von Kette B.

Falls nach einer Manipulation auf dem Mobiltelefon Guthaben an einem anderen POS ausgegeben wird, so wird dies nicht unmittelbar erkannt, sondern erst, wenn diese Transaktion über das Schwarm-Netzwerk an den Server übermittelt werden konnte. Jeder Nutzer besitzt daher ein Zugriffs-Token mit einer beschränken Gültigkeit. Ein POS lehnt einen Nutzer ohne gültiges Zugriffs-Token immer ab. Dies zwingt den Nutzer von Zeit zu Zeit online zu gehen und mit dem Server zu kommunizieren und seine Transaktionsketten mit jenen auf dem Server zu synchronisieren.

Scoop Schwarm-Netzwerk

Für die Realisierung des Schwarm-Netzwerkes wird davon ausgegangen, dass die Mobiltelefone und der POS während des Bezahlvorgangs offline sind, dass aber mindestens eines der Mobiltelefone, welches Zahlungen am POS ausführt, früher oder später eine Internetverbindung haben wird. Der POS nutzt dies nun aus und überliefert offene Nachrichten an jedes vorbeikommende Mobiltelefon. Sobald ein Mobiltelefon Internetzugriff hat, überliefert es die gespeicherten Nachrichten an den Server, welcher dann eine Empfangsbestätigung an alle Mobiltelefone ausliefert, sobald sich diese mit ihm verbinden. Mobiltelefone können nun diese Bestätigung an den entsprechenden POS ausliefern, sobald sie mit ihm interagieren. Sowohl die Mobiltelefone wie auch die POS liefern die Nachrichten solange wiederholt aus, bis sie eine entsprechende Bestätigung erhalten haben. Analog können Meldungen vom Server zum POS übertragen werden.

Die Struktur dieses Schwarm-Netzwerkes ist in Abbildung 3 dargestellt. Der Server (links) kommuniziert mit den Mobiltelefonen (Mitte), welche Nachrichten an die POS (rechts) weiterleiten bzw. Meldungen der POS ausliefern können. Der in Abbildung 3 dargestellte Graph ist bipartit und besteht aus den beiden Teilmengen der Mobiltelefone und der POS. Der Server kann dabei als spezieller POS angesehen werden. Das Schwarm-Netzwerk erlaubt, dass neben Mobiltelefonen auch Karten am System partizipieren können.

Bei einer naiven Implementierung kann dies zu einer grossen Flut an Nachrichten führen. Durch Einbezug der Position, der Internetverfügbarkeit der Mobiltelefone und des Kaufverhaltens (an welchen POS ein Mobiltelefon in der Vergangenheit Transaktionen ausgeführt hat) kann die Auslieferung von Nachrichten auf wenige Mobiltelefone beschränkt werden.

Scoop Simulation

Um ein Gefühl zu bekommen, wie schnell Meldungen beim Server ankommen, wenn sie über das Schwarm-Netzwerk verschickt werden, haben wir einen Simulator entwickelt. Mit Hilfe einer domänen-spezifische Sprache (DSL) kann die Anzahl POS und Kunden sowie deren Kaufverhalten definiert werden. Zudem kann definiert werden, wie viele POS in einer Offline-Zone stehen, in der Kunden keinen Internetzugriff haben, sowie wann und wie oft ein Mobiltelefon online ist. Basierend auf diesen Parametern wird dann eine diskrete Ereignissimulation über einen bestimmten Zeitraum durchgeführt und die Resultate gesammelt und aggregiert.

Eine Simulation unter eher pessimistischen Annahmen (25% der POS in einer Offline-Zone, sowie 33% der Mobiltelefone immer offline) mit insgesamt 100 POS und 3750 Kunden hat ergeben, dass die durchschnittliche Nachricht vom POS in-

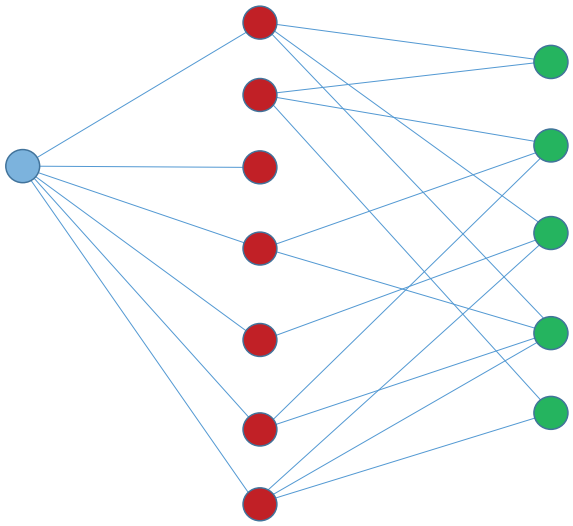


Abbildung 3: Struktur des Schwarm-Netzwerkes: Server (links), Mobiltelefone (Mitte), POS (rechts)

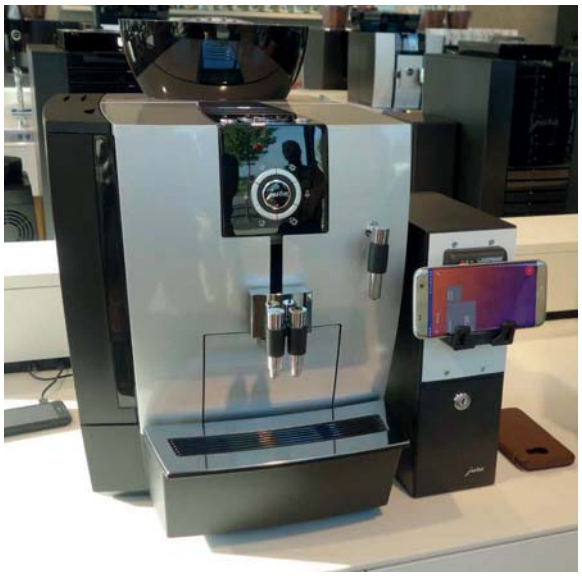


Abbildung 4: Prototypinstallation der SCOOP Paymentlösung in einem Jura Kaffeeautomaten

nerhalb von 35 Sekunden beim Server ankommt und dass die Mobiltelefone maximal 45 Nachrichten gespeichert haben, die auf eine Auslieferung warten. Beim POS waren es sogar nur maximal 8 Nachrichten, die unbestätigt auf eine Auslieferung warteten. Da die Nachrichten klein sind, ist diese Last kaum bemerkbar.

Resultat

Im Rahmen unseres Projektes haben wir das Scoop-Protokoll umgesetzt. Dazu gehört die SE-Applikation, welche die Ketten auf dem POS verwaltet und welche für die Sicherheit verantwortlich ist. Dieses SE wird vom Controller aus angesprochen, der sich im POS befindet. Dieser Controller ist auch für die Kommunikation mit den Mobiltelefonen über NFC sowie für die Implementierung des Schwarm-Netzwerkes zuständig. Auf den Mobiltelefonen ist neben der eigentlichen Scoop-Purse-Applikation auch ein *Host Card Emulation* (HCE)-Service aktiv, mit dem eine NFC-Karte emuliert wird. Das Mobiltelefon kommuniziert schlussendlich via Internet mit dem Server.

Als Resultat existiert ein erster Prototyp, der das Bezahlen von Kaffees aus einem *Jura* Kaffeeautomaten unterstützt (vgl. Abb. 4). In einer nächsten Phase werden im Rahmen einer Pilot-Installation in einer Firma sämtliche POS (Kaffee- und Snack-Automaten sowie Mensa) mit Scoop ausgerüstet.

Ein Nachteil unserer Lösung ist, dass diese aktuell nur mit Mobiltelefonen funktioniert, welche eine NFC-Schnittstelle besitzen, d.h. es funktioniert nur mit Android-Geräten. Es ist geplant, diese Einschränkung in einem weiteren Projekt anzugehen, denn wir rechnen nicht damit, dass Apple den Zugriff auf die NFC-Schnittstelle in absehbarer Zeit vollständig freigeben wird.

Referenzen

[Mau09] David Maurer; Einblicke in die Ökonomie der Zahlungskartensysteme, SNB 2009: https://www.snb.ch/de/mmr/reference/Zahlungskarten/source/Zahlungskarten.de.pdf

[Stau17] Anita Staudacher; Aus für „Quick" sorgt für Probleme: https://kurier.at/wirtschaft/aus-fuer-quick-sorgt-fuer-probleme/278.600.942

[Sabba16] Yaasha Sabba and Jordan Scheinfield; Token check offline, US Patent 20160224977: https://patents.google.com/patent/US20160224977

[iBeam14] iBeam: Datenaustausch via NFC, Projekt 20130331-05_iBeam, gefördert durch den Forschungsfonds Aargau.

[AGKW16] C. Arnosti, D. Gruntz, M. Knecht, S. Wullschleger, System zum offline-Bezahlen mit E-Geld mit mobilem Gerät mit kurzer Transaktionszeit und abschliessendem Settlement, EP-Patentanmeldung Nr. 16205267.4-1958, 2016

Links

https://de.wikipedia.org/wiki/Cash_(Geldkarte)

https://de.wikipedia.org/wiki/Quick_(Geldkarte)

https://de.wikipedia.org/wiki/GeldKarte

https://swissbit.com/de/products/security-products/micro-sd-memory-cards/ps-100u-ve

BACnet/IT – Gebäudeautomation in Zeiten des Internets

Über die letzten Jahrzehnte hat Datenkommunikation in verschiedenen Anwendungsbereichen Einzug gehalten und zu jeweils eigenen Lösungen geführt. So haben die Gebäudeautomations-Ingenieure für ihre Zwecke verschiedene Datennetzwerkstandards entwickelt, die sich in der Industrie und in bestehenden Gebäuden etabliert haben. Zugleich hat sich aus verschiedenen Anwendungen wie der Datenverarbeitung und der Büroinformation die bekannte Internet-Technologie entwickelt. Diese ist mittlerweile so verbreitet, dass eine Sogwirkung einsetzt: Dank hoher Stückzahlen, Marktmacht, Bekanntheit usw. wird es immer interessanter die Internet-Standards anstelle eigener, anwendungsspezifischer zu nutzen. Wie man zwei etablierte Standards – BACnet aus der Gebäudeautomation und die IT-Netze der Gebäude-Nutzer – zusammenbringen kann, ohne dass sich eine der beiden Welten komplett der anderen unterwirft, haben wir in einem KTI-geförderten Projekt zur Konvergenz der Gebäudeautomation und IT-Welt untersucht.

Wolfgang Weck | wolfgang.weck@fhnw.ch

Digitalisierung, *Industrie 4.0* und *Internet of Things* (IoT) sind aktuell vielgebrauchte Schlagworte, mit denen Fortschritt aber auch Handlungs- und vor allem Forschungsbedarf konnotiert sind. Die Herausforderung dabei ist häufig der Brückenschlag zwischen der Internet-Technologie der Informatik und anderen Ingenieursdisziplinen. Das *Internet der Dinge* meint ja im Kern nichts anderes als in ihrer Art meist bereits existierende technische Geräte – oft Sensoren oder Aktoren – neu über das Kommunikationsmedium Internet so zu verbinden, dass sie direkt Daten und Kommandos austauschen können, aber auch für Datenanalysedienste oder optimierende Steuerungen erreichbar werden. Dafür stehen dann wiederum die ebenfalls aktuellen „Smart"-Schlagworte wie *Smart City*, *Smart Campus*, *Smart Grid* usw.

Vor diesem Hintergrund haben wir uns in einem kürzlich abgeschlossenen KTI-Projekt[1] zusammen mit der *Siemens Building Technologies Division* in Zug mit der Konvergenz von Technik und Methodik der Informatik mit solchen aus der Gebäudeautomation beschäftigt. Dabei ging es um den Betrieb grosser Gebäude u.a. mit komplexen Lüftungs-, Heizungs- und Klimatisierungsanlagen; nicht zu verwechseln mit den seit einiger Zeit auf den Markt drängen Produkten der Heim-Automation – also Anwendungen in Wohnungen und Einfamilienhäusern hauptsächlich zur Steuerung von Licht, Beschattung und Unterhaltungselektronik.

Hier berichten wir über diese Entwicklung hin zu einer gemeinsamen technischen Basis für die zwei etablierten Ingenieursdisziplinen Gebäudeautomation und Informatik. Im Sinne eines Brückenschlags zwischen diesen Disziplinen setzt dieser Bericht keine vertieften Kenntnisse voraus – weder in Gebäudeautomation noch in Informatik.

Kurze Geschichte digitaler Datenkommunikation

Die zuvor erwähnten neuen Produkte für Heimanwendungen mögen den Eindruck vermitteln, dass Digitalisierung und IoT für die Gebäudeautomation eine ganz neue Entwicklung seien. Dass dies nicht der Fall ist, zeigt Abbildung 1 in der die zeitliche Entwicklungsgeschichte zweier Datenkommunikations-Standards gegenübergestellt ist. Die Geschichte dieser beiden Datenkommunikations-Standards beginnt bereits Ende der 80er Jahre, nachdem die Entwicklung von Elektronik und Mikroprozessoren dafür gesorgt hatte, dass Werkzeuge zur Verarbeitung digitaler Daten und Signale grundsätzlich zur Verfügung standen.

Verschiedene Nutzergruppen machten sich daran, für ihre Zwecke nutzbare Anwendungen auf dieser Basis zu kreieren. So entstanden im Wesentlichen parallel und voneinander unabhängig Datenkommunikationsstandards z.B. in der Gebäudeautomation (im Beispiel *BACnet* [BACnet], seit 2003 ISO-Standard [BACISO]) und für die Verteilung elektronischer Dokumente zunächst in der

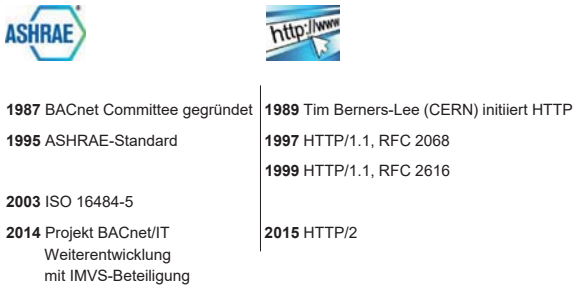


Abbildung 1: Zeitliche Entwicklung zweier digitaler Daten-Kommunikationsstandards im Vergleich: BACnet (Gebäudeautomation) und HTTP (Internet)

1 KTI-Projekt: Convergeence of Building Automation and IT World, KTI-Nr. 16841.1 PFES-ES

Wissenschaft und später in der Büro-Automation (sog. *IT-Netze*). Erst über die folgenden Jahre bis Dekaden entwickelte sich aus Letzterem rund um die Protokolle TCP, HTTP usw. sowie die dafür genutzten Kabel- und Funkstandards so viel Dynamik und Marktmacht, dass sich daraus ein praktisch flächendeckend vorhandener Service etablierte – das *Internet* aus Konsumentensicht. Den Zugang dazu betrachtet man heute vielfach als ähnlich selbstverständlich vorhanden wie den zu elektrischer Energie und trinkbarem Wasser.

Die Datenkommunikationsstandards der Automatisierung, wie das in Abbildung 1 gezeigte BACnet, umfassten zunächst separate Entwicklungen auf allen relevanten Ebenen von der Verkabelung bis zu den anwendungsspezifischen Datenformaten für Sensor-Messwerte und Steuerbefehle [Bus97]. Von der Dynamik der Entwicklungen der IT-Netze blieben sie mehr oder weniger unberührt. Einerseits bleiben bestehende Installationen in Gebäuden lange Zeit erhalten, da es aufwändig aber wenig nutzbringend wäre sie zu ersetzen. Andererseits gab es wenig Druck zu Veränderungen auf die Gebäudeautomation. Die bestehenden Standards konnten gut weiterentwickelt werden und teilweise auch auf industriell breit gefertigte Standardhardware übertragen werden. So sind die Kabelstandards neuerer BACnet-Anlagen identisch mit jenen der IT-Netze, die Datenübertragungsformate aber weiterhin unverändert, was die Integration neuer Geräte in bestehende Gebäude – z.B. beim Ersatz nach Defekt – möglich macht.

Die Digitalisierung hat sich also in der Gebäudeautomation und der Informatik parallel entwickelt. Dies ermöglichte auch, unterschiedliche Anforderungen zu unterstützen. Während es beim weltweit offenen Internet wichtig wurde, die eigenen lokalen IT-Netze (LAN oder Intranet) und Geräte vor Missbrauch durch Datenkommunikation von aussen zu sichern, war es für die physikalisch ohnehin von der Aussenwelt getrennten Gebäudeautomationsnetze vor allem wichtig, innerhalb des Gebäudes Daten direkt und schnell – das heisst innerhalb enger Zeitfenster – zu übertragen.

Die Dichte von Geräten wird dabei im Internet der Dinge sehr hoch. Betrachten wir dies exemplarisch am Beispiel des FHNW-Standorts Windisch, wo sich auch die Hochschule für Technik mit dem IMVS befindet. Hier wird zwar nicht BACnet verwendet, sondern mit KNX ein anderer Kommunikationsstandard der Gebäudeautomation [KNX]. Zwar ist KNX – wie BACnet – ein anwendungsbezogener Kommunikationsstandard und das Netz ist a priori nicht direkt mit dem Internet verbunden. Die Perspektive des IoT ist aber durchaus, das zu ändern. Es ist auch heute schon vergleichsweise einfach, einen kleinen preisgünstigen Computer (z.B. einen Raspberry Pi [RasPi]) gleichzeitig mit dem KNX-Bus und dem Internet zu verbinden und so einen offenen Zugang zur Gebäudeautomation zu schaffen.

Am KNX-Bus des Standorts Windisch der FHNW sind ca. 13000 Sensoren und Aktoren angeschlossen. Zum Vergleich: Die Anzahl von Computern und sonstigen Geräten (Server, Drucker, etc.), die mit dem IT-Netz verbunden sind, beträgt an der FHNW in Windisch ca. 3500, also weniger als ein Drittel. Erst für alle Standorte der FHNW zusammengenommen kommt die Anzahl der Geräte im IT-Netz mit 12000 in eine ähnliche Grössenordnung wie diejenige der „Dinge im Gebäudeautomationsnetz“ alleine in Windisch.

BACnet/IT – Konvergenz von Gebäudeautomation und Internet

Im Rahmen des oben erwähnten KTI-Projekts wurde auf Initiative des BACnet-Standardkomitees [BACnet] die Möglichkeiten zur Konvergenz der beiden parallelen Entwicklungen von IT-Netzen und BACnet untersucht. Kurzgefasst kann man das Ziel der Untersuchungen formulieren als „ein Netz für alles“. Das heisst, die Gebäudeautomation soll die ohnehin vorhandene Internet-Infrastruktur mitbenutzen: Nicht nur die gleichen, sondern dieselben Kabel und Services wie das IT-Netz für Büroautomation, Unterhaltungselektronik und das Internet.

Hinter diesem Ziel stecken zwei treibende Kräfte. Erstens hat sich die IT-Netzwerk-Technologie heute soweit ausgebreitet, dass entsprechende Installationen als vorhanden bzw. ohnehin notwendig vorausgesetzt werden können, analog zu denen für die Verteilung elektrischer Energie. Die nötigen Geräte samt der zum Betrieb relevanter Dienste nötigen Software werden industriell in grossen Stückzahlen preisgünstig gefertigt. Es ist also mit substanziellen Kostenreduktionen zu rechnen, wenn man bei neuen Gebäuden auf die Internet-Technologie setzen kann, anstatt eine komplette Parallelwelt aufzubauen.

Nur ein Netz anstelle von mehreren verschiedenen zu betreiben, macht es ausserdem leichter, die Datenkommunikation zuverlässiger zu machen. Investiert man in redundante Einrichtungen dieses einen Netzes, kommt das gleichzeitig allen Netznutzern zugute, kostet aber nur einmal.

Die zweite treibende Kraft ist die immer wichtigere und nutzbringendere Öffnung der Kommunikation von lokalen Anlagen, also der direkten Verbindung mit dem Internet. Sei es, dass man Cloud Services zur Erfassung grosser Datenmengen aus Gebäuden nutzen möchte, um sie zu analysieren und damit beispielsweise den Energieverbrauch eines Gebäudes optimieren zu können; sei es, dass man in die Steuerung von aussen eingreifen können möchte, um auch externe Informationsquellen wie Wetter- oder Energiepreisprognosen einfliessen zu lassen; sei es, dass man ein-

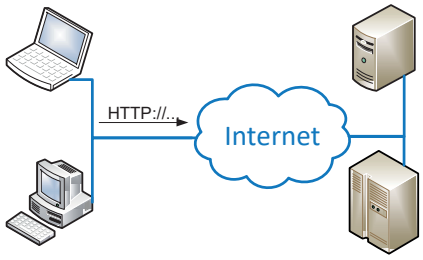


Abbildung 2: Typischer Verbindungsaufbau vom Client zum Server in IT-Netzen und im Internet

fach Daten und Steuerung über mehrere Gebäude hinweg verbinden möchte (*Smart Campus*).

Herausforderungen

In unserem Forschungsprojekt zeigten sich zwei Hauptherausforderungen, die wir hier vertieft betrachtet wollen. Diese Herausforderungen ergeben sich gerade aus den bisher unabhängigen Entwicklungen, Anforderungen und eben auch Kulturen der Gebäudeautomations-Ingenieure und der Informatiker. Kurzgefasst handelt es sich um folgende zwei Punkte:

1. Aus Sicherheitsgründen segmentierte IT-Netze behindern die Peer-to-Peer-Kommunikation im IoT. Die Herausforderung besteht darin, Verbindungen zwischen beliebigen IoT-Geräten möglich zu machen, was professionelles IT-Netzwerkmanagement unter Umständen gerade zu verhindern versucht.
2. Die für sich abgeschlossenen Netze der Gebäudeautomation sind nicht auf die Sicherheitsrisiken vorbereitet, die sich durch eine Verbindung mit dem Internet ergeben. Die Herausforderung besteht darin, jegliche Datenkommunikation eines Geräts nur mit erkennbar berechtigten Kommunikationspartnern zuzulassen, seien diese innerhalb oder ausserhalb des gleichen Gebäudes.

Sicherheit in IT-Netzen behindert IoT

IT-Netze werden meist für Client-Server-Architekturen konfiguriert (s. Abb. 2). Arbeitsplatz-Computer und mobile Geräte sind Clients, die Verbindungen zu Servern aufbauen, um deren Services zu nutzen. Im einfachsten Fall ist das eine Webseite, die vom Server als Antwort auf eine Anfrage geliefert und dann vom Browser-Programm auf dem Client dargestellt wird. Ein Client kann aber auch ein Programm sein, das einen Microservice beansprucht, der von einem Server angeboten wird. Dabei spielt es a priori keine Rolle, ob sich der Server innerhalb des gleichen IT-Netzes befindet oder ausserhalb, z.B. als sogenannter Cloud-Server bei einem entsprechenden Anbieter.

Die Kommunikation zwischen Clients und Servern ist üblicherweise verbindungsorientiert (basierend auf dem TCP-Protokoll). Nur über eine vom Client her eröffnete Verbindung kann ein Server seine Antwort schicken. Man möchte hingegen nicht, dass eine Verbindung hin zu einem der Clients eröffnet wird. Das wäre nicht konform mit dem Client-Server-Ansatz und möglicherweise der Versuch eines böswilligen Angriffs.

Das IT-Netz der FHNW beispielsweise verbindet eine grosse Menge an persönlichen Clients von Studierenden und Mitarbeitenden. Die meisten dieser Geräte werden auch ausserhalb der Hochschule mitgenommen und mit anderen Netzen verbunden. Da ist es schnell möglich, dass ein solches Gerät, das irgendwo anders einmal von Schadsoftware infiziert wurde, im FHNW-Netz versucht, andere Clients ebenfalls zu infizieren. Um das zu verunmöglichen, ist das logische Teilnetz, in dem sich alle diese Clients befinden, so konfiguriert, dass überhaupt keine Verbindungen zu diesen Teilnehmern eröffnet werden können – weder von einem anderen Gerät innerhalb desselben Teilnetzes, noch von ausserhalb.

Die IT-Server und Services eines Unternehmens lassen sich grob in zwei Gruppen teilen: Solche die man der Öffentlichkeit über das Internet anbieten möchte, wie Informationsseiten, Kundenportale, E-Mail-Empfang usw., und solche, die interne Dienste erbringen, die nur für Mitarbeitende des Unternehmens zur Verfügung stehen dürfen. Letzteres sind Datenbanken, Verwaltungssysteme und meist alles, was die eigentliche Geschäftstätigkeit ausmacht.

Der Datenkommunikationszugang von aussen wird deswegen meist zweistufig gesichert (s. Abb. 3). In einem oft als „demilitarisierte Zone“ (DMZ) bezeichneten Teilnetz sind Server erreichbar, deren Dienste im Internet öffentlich zugänglich sein sollen. Eine Firewall blockiert dabei Versuche, zu

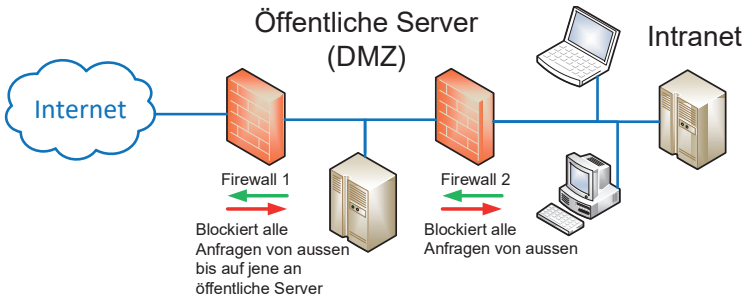


Abbildung 3: Zweistufige Sicherung des internen Netzwerks mit Firewalls

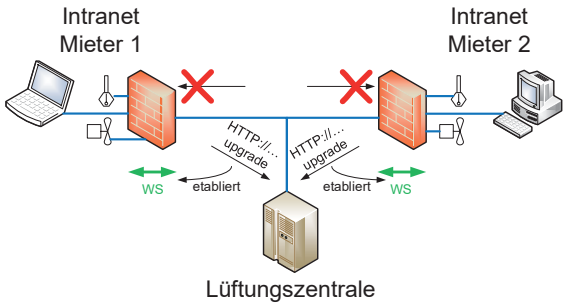


Abbildung 4: Bidirektionale Datenkommunikation auch in Mieternetze dank Websockets (WS)

diesen Maschinen unerwünschte Verbindungen zu eröffnen, also solche, die nicht direkt einem publizierten Service zugeordnet werden können. Oft wird daher nur das HTTP bzw. HTTPS-Protokoll zugelassen. Eine zweite Firewall verbietet jegliche Verbindungseröffnung zu den Geräten im internen Netzwerk. Diese dürfen – ganz gemäss Client-Server-Modell – Verbindungen aus diesem inneren Netz nach aussen öffnen, aber nicht umgekehrt.

Im Gegensatz zu Client-Server-Architekturen benutzt Gebäudeautomation traditionell ein verbindungsloses Kommunikationsmodell (wie das UDP-Protokoll). Anfragen nach Messwerten von Sensoren und Stellkommandos an Aktoren sowie Antworten und Bestätigungen werden einfach als einzelne Nachrichtenpakete im Netz verschickt. So kann beispielsweise ein Raumthermostat direkt einem Heizungsventil Kommandos zum Öffnen oder Schliessen zustellen. Die Firewalls der IT-Netze blockieren solche (UDP-)Nachrichten, die nicht über akzeptierte TCP-Verbindungen verschickt werden im Allgemeinen.

Stellen wir uns ein grosses Gebäude mit verschiedenen, sehr unterschiedlichen Mietern vor. Ein solcher Mieter ist vielleicht eine Bank, die hohe Sicherheitsstandards an ihre IT-Netze anlegt und Personal beschäftigt, das sich aktiv um deren Einhaltung kümmert, indem es entsprechende Einrichtungen wie Firewalls betreibt und aktiv konfiguriert. Ein anderer Mieter ist vielleicht eine kleine mittelständische Firma mit wenigen Mitarbeitenden, die alle einen Laptop mit Internet-Zugang benutzen wollen, aber keinerlei eigene Server vor Ort betreiben. (Entsprechende Services laufen vielleicht extern „in der Cloud“ bzw. bei einem Service Provider). Diese Firma hat dann vielleicht einen einfachen Router mit integrierter Firewall für den Internetzugang der Mitarbeiter-Laptops installiert. Verbindungsaufbauversuche von ausserhalb des (einfachen und nicht aktiv administrierten) Firmennetzes werden typischerweise so abgewiesen.

Schwierig wird es nun, wenn z.B. eine Lüftungszentrale im Gebäude Steuerbefehle an Ventile geben muss, die sich verteilt im Gebäude befinden und damit in den individuellen IT-Netzen der verschiedenen Mieter (vgl. Abb. 4). Der bei BACnet übliche Ansatz, einfach je ein Nachrichtenpaket von der Zentrale an die betroffenen Ventile zu verschicken, wird nicht mehr funktionieren, weil die Firewalls der Mieter im Wege stehen und diese Nachrichtenpakete nicht weiterleiten werden. Denkbar wäre natürlich, alle Mieter zu bitten bzw. zu verpflichten, entsprechende Sonderregeln für ihre Firewalls und Router zu konfigurieren. Die ICT-Spezialisten der Bank aus unseren obigen Beispielszenarien wären wohl grundsätzlich in der Lage dazu. Allerdings widerspricht ein solcher Wunsch meist der internen Policy der Firma. Ausserdem kann durch die schiere Anzahl an Gebäudeautomationsgeräten die Umsetzung aufwändig werden. Bei der mittelständischen Firma wiederum gibt es möglicherweise keine Person, die sich mit der IT-Technik genügend gut auskennt, um den (einfachen) Router entsprechend zu konfigurieren.

Logisches Netz aus Websocket-Verbindungen

Es wäre nun wenig hilfreich die Erfahrungen und Methoden samt existierender Programme der Gebäudeautomations-Ingenieure auf den Client-Server-Stil der Informatik zu zwingen, oder – umgekehrt – von IT-Netzen grundsätzliche die Durchlässigkeit von einzelnen Nachrichtenpaketen zu verlangen, wie sie die Gebäudeautomation verwendet. Vielmehr braucht es einen gemeinsamen Weg, mit dem einerseits existierende Konzepte der Gebäudeautomation weiterhin genutzt werden können, andererseits IT-Netze nicht speziell konfiguriert werden müssen.

Hier kommt ein Werkzeug zum Zug, das die Informatik ursprünglich entwickelt hat, um manche Grenzen der Client-Server-Architektur zu überwinden. In der ursprünglichen einfachen Form dieser Architektur muss nämlich jeder Client, der z.B. aktuelle Daten von einem Server benötigt, den Server immer wieder neu nach Veränderungen der Daten anfragen. Reaktionsschneller ist es aber, wenn der Server den Client von sich aus über solche Änderungen informieren kann. Dazu müssen Client und Server gemeinsam die am Anfang der Kommunikation etablierte Netzverbindung offen behalten, damit so der Server immer wieder neue Daten an den Client schicken kann. Dafür hat sich unter dem Namen *Websocket* ein entsprechender Standard etabliert [FM11].

Websockets sind (TCP-)Verbindungen, die sich mittels üblicher HTTP-Anfragen etablieren und nachher nahezu uneingeschränkt in beiden Richtungen nutzen lassen. So können sie auch in Netzen mit Firewalls benutzt werden, weil die Eröffnung dem Client-Server-Ansatz entspricht: Die Verbindung wird aus einem internen Netz heraus wie von einem Client bestellt.

Abbildung 4 zeigt, wie eine Lüftungszentrale des Gebäudes Ventile steuern kann, die über die internen IT-Netze der Mieter angeschlossen sind: Jedes Ventil verbindet sich einmalig mit einem Server bei der Zentrale und hält diese Verbindung offen – als Websocket. So kann die Zentrale es jederzeit für Kommandos erreichen. Wird die Verbindung aus irgendwelchen Gründen einmal unterbrochen, müssen die Kommunikationspartner dies feststellen und dann kann das Ventil eine neue Verbindung öffnen, um die bisherige zu ersetzen.

In einem Gebäude mit BACnet über IT-Netze verbinden sich also zunächst die einzelnen BACnet-Geräte untereinander indem sie ein Netz von Websockets etablieren über das sie dann im Bedarfsfall ihre Nachrichten nach bewährtem BACnet-Schema austauschen. Dass die Websockets nicht erst geöffnet werden, wenn konkret eine Nachricht übertragen werden muss dient auch dazu, Zeitverluste beim Nachrichtenversand zu vermeiden.

Dieses logische Netz aus Websocket-Verbindungen kann nach Bedarf aus einer Reihe direkter Verbindungen zwischen verschiedenen BACnet Geräten bestehen, aber auch als Stern mit einem zentralen Hub oder Broker organisiert werden. Direkte Verbindungen sind vor allem bei sehr zeitkritischen Anwendungen von Feuermelder bis Lichtschalter nützlich. Andererseits vereinfacht eine Sterntopologie die Verwaltung der Verbindungen. Möglich sind beide Ansätze.

Security – Datenkommunikation auf berechtigte Partner einschränken

Die Verwendung von Websockets unterläuft nun – zunächst ja beabsichtigt – die Sicherheitsmechanismen der IT-Netze. Gleichzeitig können sie Geräte der Gebäudeautomation mit dem öffentlichen Internet verbinden. Diese Mischung trägt grosses Gefahrenpotenzial in sich, denn es ist nun denkbar, dass irgendjemand von einem beliebigen Ort der Welt aus mit einzelnen Geräten eines Gebäudes Kontakt aufnehmen und Messwerte lesen oder Steuerkommandos geben kann.

Um dies zu verhindern, müssen die folgenden zwei Hauptanforderungen an die Sicherheit eines Gebäudeautomations-Netzes gestellt werden:

1. Unberechtigten darf es nicht möglich sein, Datenverkehr mitzulesen. Daraus liessen sich z.B. Erkenntnisse gewinnen, wann Räume leer stehen, um Einbrüche zu planen.
2. Geräte müssen erkennen können, ob Kommandos, die sie erhalten, aus einer Quelle stammen, die zur gleichen Anlage gehört.

Das heisst also, die Datenübertragung muss verschlüsselt werden (Kryptographie) und die Identität des Absenders einer Nachricht muss überprüfbar sein (Authentifizierung).

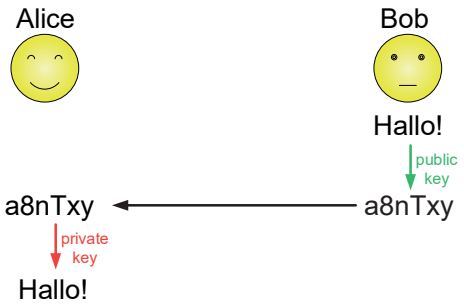


Abbildung 5: Verschlüsselte Datenübertragung beliebiger Sender (hier Bob), die nur mit privatem Schlüssel entschlüsselt werden kann

Public Key Cryptography

In der IT-Welt etablierte Lösungen für die beiden zuvor genannten Aufgaben beruhen auf asymmetrischer Public Key Cryptography. Diese arbeitet mit Paaren von kryptographischen Schlüsseln – z.B. S_1 und S_2. Jeder solcher Schlüssel ist eine mathematische Funktion, die aus einer Folge von Bytes eine andere Byte-Folge erzeugt. Schlüssel S_1 angewandt auf eine Nachricht N – also $S_1(N)$ – liefert eine verschlüsselte Form der Nachricht N.

Speziell an asymmetrischer Kryptographie ist, dass man eine Nachricht (bzw. Byte-Folge) die mit einem der beiden Schlüssel S_1 oder S_2 verschlüsselt worden ist, nur mit dem jeweils anderen Schlüssel wieder entschlüsseln kann. Es gilt also:

$S_2(S_1(N)) = N$ bzw. $S_1(S_2(N)) = N$.

Wer nur einen der beiden Schlüssel kennt, kann zwar verschlüsseln aber die gerade verschlüsselte Nachricht nicht wieder entschlüsseln.

In der Anwendung erklärt man nun einen der beiden Schlüssel als öffentlich (oder eben *public*) und den anderen als *privat*. Der private Schlüssel darf nur auf einem einzigen Gerät bekannt sein und muss dort geheim gehalten werden. Er wird also nie über das Netz verschickt. Den öffentlichen Schlüssel hingegen darf jeder und jedes beliebige Gerät kennen.

Zwei Geräte, die verschlüsselt miteinander kommunizieren sollen, können also einfach zu Beginn ganz offen ihre jeweiligen öffentlichen Schlüssel austauschen. Dieser Austausch darf auch von böswilligen Gegnern mitgehört werden, denn mit diesen Schlüsseln alleine ist keine Entschlüsselung möglich. Ist dies einmal geschehen, kann jedes Gerät den öffentlichen Schlüssel des Kommunikationspartners verwenden, um diesem Nachrichten so verschlüsselt zu schicken, dass nur er sie wieder entschlüsseln kann (s. Abb. 5).

Die Kosten dieser Public Key Cryptography fallen vor allem in Form von Rechenaufwand beim Ver- und Entschlüsseln an. Dieser Rechenaufwand ist höher, als bei einfacheren symmetrischen Verschlüsselungsverfahren, was gerade bei sehr Ressourcen-beschränkten Geräten im IoT-Bereich problematisch sein kann.

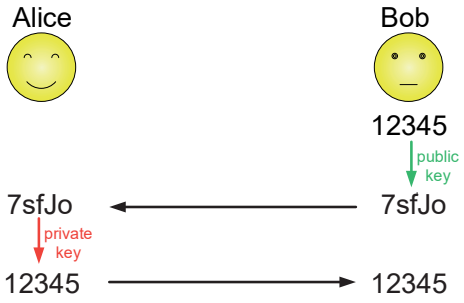


Abbildung 6: Authentifizierung als Besitzer des privaten Schlüssels zu einem öffentlichen Schlüssel mittels «Challenge Response»-Verfahrens

Symmetrische Verschlüsselung beruht auf je einem Schlüssel, den sowohl Sender als auch Empfänger einer Nachricht kennen. Treten zwei Geräte in Kontakt, müssen sie Information austauschen, die ausreicht, um diesen Schlüssel auf beiden Seiten identisch zu berechnen. Diese Information könnte aber auch ein Angreifer abhören, der dann ebenfalls den Schlüssel kennen und anschliessend abgehörte verschlüsselte Nachrichten entschlüsseln könnte.

Für verbindungsorientierte Kommunikation wie für Client-Server-Architekturen verwendet man in der Informatik eine Kombination dieser beiden Ansätze. Zunächst wird Public Key Cryptography benutzt, um die Möglichkeit zu schaffen, überhaupt sicher verschlüsselt zu kommunizieren. Dieses aufwändige Verfahren dient somit dazu – abhörsicher – einen symmetrischen Schlüssel zu vereinbaren, der anschliessend – recheneffizient – für den eigentlichen Austausch grösserer Datenmengen verwendet wird.

Ein dafür geeignetes Ablaufprotokoll ist unter Namen wie *TLS* oder *Secure Websockets* standardisiert und es existieren Implementierungen in Form von Bibliotheken, die in eigene Programme eingebunden werden können.

Authentifizierung mit X.509-Zertifikaten

Public Key Cryptography lässt sich auch als Werkzeug nutzen, um das zweite Sicherheitsproblem zu lösen. Es nutzt ja nichts, verschlüsselt zu kommunizieren, wenn ein Teilnehmer unbeabsichtigt und unwissentlich seine Nachrichten einem falschen Empfänger schickt und zur Verschlüsselung dessen öffentlichen Schlüssel benutzt. Der Sender sollte daher sicher sein, auch den richtigen Empfänger zu erreichen.

Der öffentliche Schlüssel wird dazu als Teil eines *Zertifikats* übermittelt, das die Funktion eines *Ausweises* hat. Analog zum Reisepass muss ein solches Zertifikat zwei Beweise möglich machen: Zum Ersten muss überprüfbar sein, dass der Ausweis für denjenigen ausgestellt ist, der ihn präsentiert (entsprechend dem Passfoto oder biometrischen Daten). Zum Zweiten muss geprüft werden können, dass die Angaben im Ausweis (z.B. Name oder Zugehörigkeit zu einer bestimmten Gemeinschaft) korrekt sind und der Ausweis von vertrauenswürdiger Stelle ausgestellt wurde.

Um zu überprüfen, ob ein Zertifikat tatsächlich vom Kommunikationspartner am anderen Ende einer Verbindung stammt und nicht einfach von diesem weitergeleitet worden ist, wird wieder der öffentliche Schlüssel benutzt. Allerdings geht es diesmal nicht darum, Daten abhörsicher zu übertragen, sondern zu überprüfen, ob das Gerät am anderen Ende einer Kommunikationsverbindung das Gegenstück zum öffentlichen Schlüssel, den privaten Schlüssel kennt. Wie Abbildung 6 zeigt, lässt sich dies sehr einfach nachprüfen: Man schickt eine beliebige zufällig gewählte Nachricht mit dem öffentlichen Schlüssel verschlüsselt an den Kommunikationspartner und lässt sich von diesem die entschlüsselte Nachricht zurückschicken. Die retournierte Nachricht ist nur identisch mit der ursprünglichen, wenn der Kommunikationspartner tatsächlich über den privaten Schlüssel verfügt, der zum verwendeten öffentlichen Schlüssel gehört.

Ein Zertifikat enthält eine Reihe von Angaben über den Kommunikationspartner. Beispielsweise seine Adresse, einen Namen und – für unsere BACnet-Anwendung – die Information, zu einer bestimmten Anlage zu gehören. Letztere ist ausschlaggebend, denn die Sicherheit von BACnet/IT baut darauf auf, dass nur Geräte derselben Anlage miteinander Verbindungen unterhalten dürfen.

Offensichtlich wäre es für einen Angreifer ein Leichtes, ein Schlüsselpaar (privater und öffentlicher Schlüssel) zu erzeugen und damit ein Zertifikat zu generieren, das aussagt, dass der Inhaber zu einer Anlage nach Wahl gehört. Wie sich aber niemand selbst einen anerkannten Reisepass ausstellen kann, sondern dafür einen vertrauenswürdigen Herausgeber benötigt, müssen auch Zertifikate von einer sogenannten *Certificate Authority* herausgegeben bzw. beglaubigt werden.

Diese Beglaubigung geschieht durch eine sogenannte *Signatur* des Zertifikats. Abbildung 7 zeigt das Schema dazu. Mit einem fixen Algorithmus (es gibt mehrere mögliche und das Zertifikat enthält den Namen des zu benutzenden Algorithmus) wird aus dem zu signierenden Zertifikat eine Art Prüfsumme (ein sogenannter Hashwert) berechnet. Das ist im Prinzip eine grosse Zahl, die man zu einem gegebenen Zertifikat rasch berechnen kann, für die es aber extrem schwierig ist, ein anderes nützliches Zertifikat zu erzeugen, das zur gleichen Prüfsumme führen würde. Diese Prüfsumme wird von der *Certificate Authority* mit deren privatem Schlüssel verschlüsselt und dem Zertifikat als „Signatur“ hinzugefügt.

Um die Echtheit eines solchen Zertifikats zu prüfen, kann jeder, der den öffentlichen Schlüs-

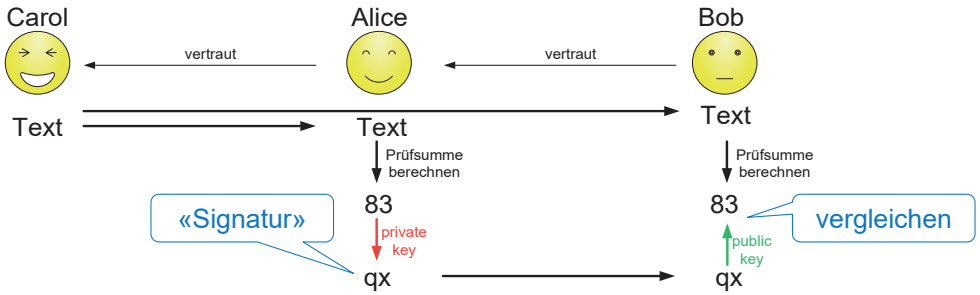


Abbildung 7: Signieren eines X.509-Zertifiakts durch Alice als vertrauenswürdige Certificate Authority: Durch Hinzufügen einer mit dem privaten Schlüssel verschlüsselten «Prüfsumme» zu einem Text von Carol gibt Alice allen ihr vertrauenden Partnern bekannt, dass sie diesem Text von Carol ebenfalls vertrauen können.

sel der *Certificate Authority* kennt, die Signatur damit entschlüsseln und mit der selbst zum Zertifikat berechneten Prüfsumme vergleichen. Sind die beiden Werte gleich, kann davon ausgegangen werden, dass das vorliegende Zertifikat mit genau diesem Inhalt von der *Certificate Authority* für vertrauenswürdig beurteilt worden ist. Mit diesem Prozess kann also Vertrauen weitergegeben werden.

Die hier erklärten Verfahren sind Grundlagentechniken der Informatik. Glücklicherweise sind die Informatiker aber über die reinen Konzepte hinausgegangen und haben z.B. Formate für Zertifikate (sogenannte X.509-Zertifikate) sowie Protokolle für den Austausch und die Überprüfung sowie für verschlüsselte Kommunikation (TLS) vereinheitlicht standardisiert und in Programmbibliotheken implementiert, die man in eigene Produkte einbinden kann.

Ergebnisse

In dem hier beschriebenen Forschungsprojekt haben wir gezeigt, dass sich aus etablierten Werkzeugen der Informatik eine sichere und genügend effiziente Datenkommunikationsgrundlage schaffen lässt, auf der etablierte Standards und Anwendungen der Gebäudeautomation aufgesetzt werden können. Aus *Secure Websockets* (der standardmässigen Kombination aus *Websockets* mit TLS) lässt sich ein logisches Netz von Verbindungen auch durch Firewalls der IT-Netze hindurch errichten.

Vorhandene BACnet-Implementierungen binden dies als neues Transportmedium (in der BACnet-Welt als *Data Link* bezeichnet) ein. Die Applikationen der Gebäudeautomation benutzen unverändert die Schnittstelle ihres BACnet-Stacks.

Im Jahr 2018 wird das BACnet-Standardisierungs-Komitee der ASHRAE einen neuen Standard-Entwurf *BACnet/SC* öffentlich zur Review aufgelegen [ASHRAE]. *SC* steht dabei für *secure connect* und beinhaltet die in diesem Artikel beschriebene sichere Datenkommunikation über *Secure Websockets*.

An diesem Konzept durften wir in den vergangenen Jahren mitarbeiten, zusammen mit Ingenieuren der *Siemens Building Technologies Devision*. Sowohl von Siemens als auch von uns ist der entstehende Standard implementiert worden, um zu zeigen, dass zwei unabhängig entstehende Implementierungen zueinander kompatibel sind. Wir haben von diesen beiden Prototypen auch nicht-funktionale Qualitätsmerkmale untersucht, allen voran das Laufzeitverhalten auch bei grösserem Nachrichtenvolumen. Die Ergebnisse dieser Arbeiten sind vom Standardisierungskomitee direkt genutzt worden, um Teillösungen zu bestätigen oder zu überarbeiten.

Wir haben aus dem Projekt Erkenntnisse gewinnen können, die sich gut auf generelle IoT-Themen auch jenseits von Gebäudeautomation verallgemeinern lassen. Je nach konkreten Anforderungsprofilen können die für BACnet erarbeiteten Lösungsstrategien direkt oder in angepasster Form übernommen werden.

Referenzen

[ASHRAE] American Society of Heating, Refrigerating and Air-Conditioning Engineers: https://www.ashrae.org/

[BACnet] BACnet – A Data Communication Protocol for Building Automation and Control Networks, Official Website of ASHRAE SSPC 135: http://www.bacnet.org/

[BACISO] Der ISO 16484-5:2017-Standard in aktueller Form: https://www.iso.org/standard/71935.html

[Bus97] Steven T. Bushby: "BACnetTM – A standard communication infrastructure for intelligent buildings", Published in Automation in Construction, Vol. 6 No. 5-6, 1997, pp. 529-540. http://www.bacnet.org/Bibliography/AIC-97/AIC1997.htm

[FM11] Fette, I. and A. Melnikov, "The WebSocket Protocol", RFC 6455, DOI 10.17487/RFC6455, December 2011. http://www.rfc-editor.org/info/rfc6455

[KNX] KNX Association: https://www.knx.org

[RasPi] Raspberry Pi: https://www.raspberrypi.org

IoT als Enabler für ein interaktives Schaufenster

Am Standort Brugg/Windisch betreibt die FHNW ein Maker Studio mit einem Schaufenster zur Präsentation von Wechselausstellungen. Um Passanten in die Wechselausstellungen einbeziehen zu können, sind lichtempfindliche Schalter am Schaufenster angebracht, mit denen die interaktiven Elemente einer Ausstellung gesteuert werden können. Diese Schalter und anderen interaktiven Elemente sind mit Technologien des Internet of Things (IoT) realisiert worden. Unter IoT wird die virtuelle Repräsentation realer, physischer Objekte verstanden, die untereinander mithilfe der Internet-Technologien kommunizieren können.

Jürg Luthiger, Robin Schoch, Christoph Stamm | juerg.luthiger@fhnw.ch

Die FHNW betreibt am Standort Brugg/Windisch ein *Maker Studio* [MakStu]. Das Maker Studio stellt Studierenden, Mitarbeitenden der FHNW und der Öffentlichkeit eine Infrastruktur aus traditionellen und digitalen Werkzeugen zur Verfügung, um kreative Ideen selber umsetzen zu können. Es ist davon auszugehen, dass im Kontext des Maker Studios spannende Projekte entstehen werden. Für diese Projekte ist eine Plattform geschaffen worden, um die Projekte ausstellen und einer breiten Öffentlichkeit präsentieren zu können. Die Ausstellungen sollen einerseits die Essenz der Maker Idee [Mak] wiedergeben und andererseits das Publikum zur Interaktion animieren. In einem Schaufenster des Maker Studios kann jeweils eine Ausstellung präsentiert werden.

Das in Abbildung 1 abgebildete Schaufenster soll mit maximal zwanzig interaktiven Elementen ausgerüstet werden, um vor allem ein jüngeres Publikum anzusprechen und ihnen auf spielerische Art und Weise einen Einblick in die Möglichkeiten der Maker Kultur geben. Da über das Schau-

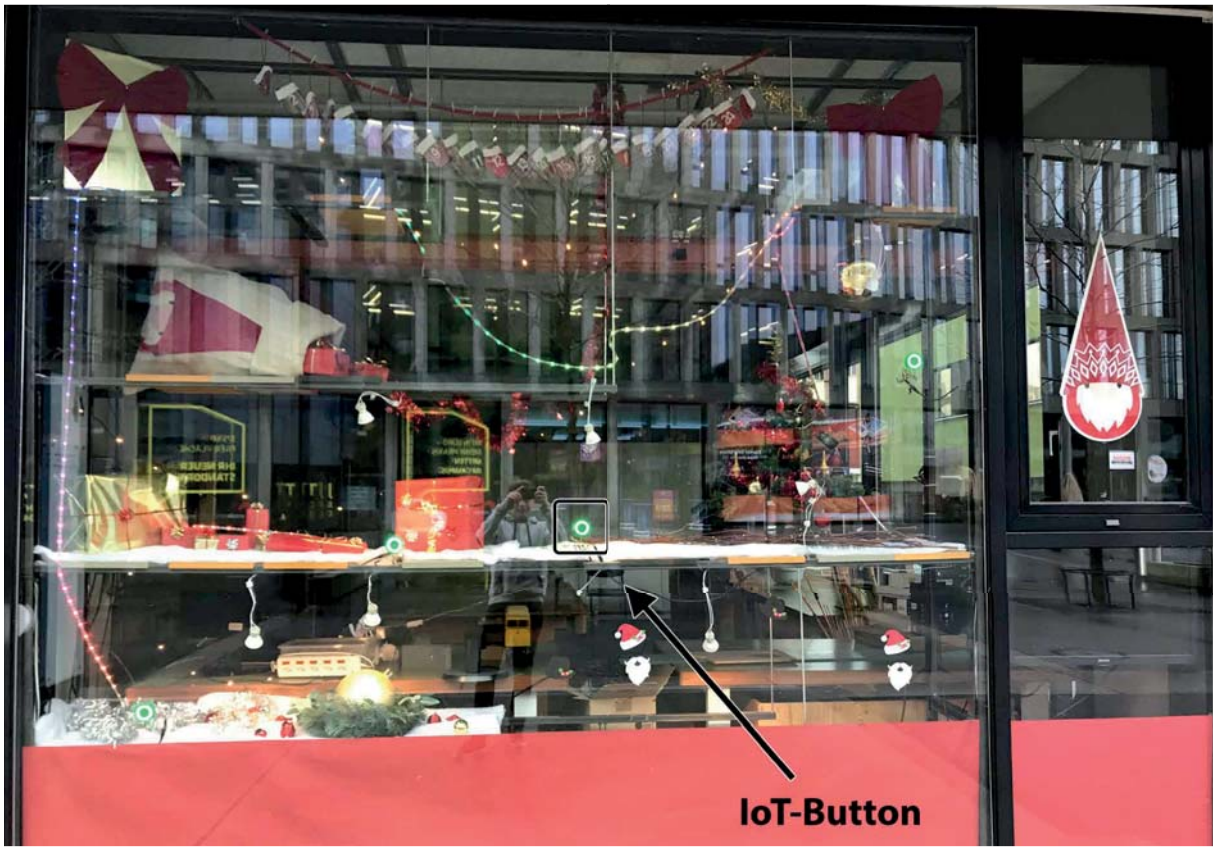


Abbildung 1: Schaufenster des Maker Studios in Brugg/Windisch

fenster wechselnde Ausstellungen gezeigt werden sollen, muss die Infrastruktur des Schaufensters dynamisch und flexibel gestaltet werden, um den Konfigurationsaufwand bei einem Ausstellungswechsel minimal halten zu können. Das gilt auch für die Verkabelung. In Anbetracht der Vielzahl der Geräte soll die Kommunikationsinfrastruktur drahtlos sein, um die Verkabelung auf ein Minimum reduzieren zu können.

Eine Wechselausstellung wird verschiedene Objekte umfassen, die mehrheitlich animiert werden können. Diese Animationen sollen aber erst durch eine Benutzerinteraktion ausgelöst werden, so dass die Benutzer in die Ausstellung involviert werden. Ebenfalls soll es möglich sein, zusätzliche Informationen zu den Objekten und zur aktuellen Ausstellung anzeigen zu können.

Das Schaufenster

Das zur Verfügung stehende Schaufenster gehört zum Werkraum, der mit traditionellen Werkmaschinen ausgerüstet ist. Das Schaufenster öffnet sich auf einen grossen Platz, wo die Passanten zirkulieren. Es zeigt in Richtung Osten, so dass der Lichteinfall am Vormittag und am Nachmittag sehr unterschiedlich sein kann. Die Werkstatt wird regelmässig benutzt und deshalb ist der natürliche Lichteinfall zu gewährleisten. Ebenfalls sind Installationen ausserhalb des Werkraums nicht möglich.

Um über das Schaufenster mit der Ausstellung interagieren zu können, ist eine geeignete Sensorik notwendig, die kostengünstig, flexibel und robust ist. Verschiedene Sensor-Technologien wie Kamera oder Näherungssensor sind evaluiert worden, aber wegen Kostenfaktor/Privatsphäre (Kamera) oder wegen hoher Signaldämpfung der Fensterfläche (Näherungssensor) wieder verworfen worden. Der Lichtsensor ist die einzige valable Alternative geblieben.

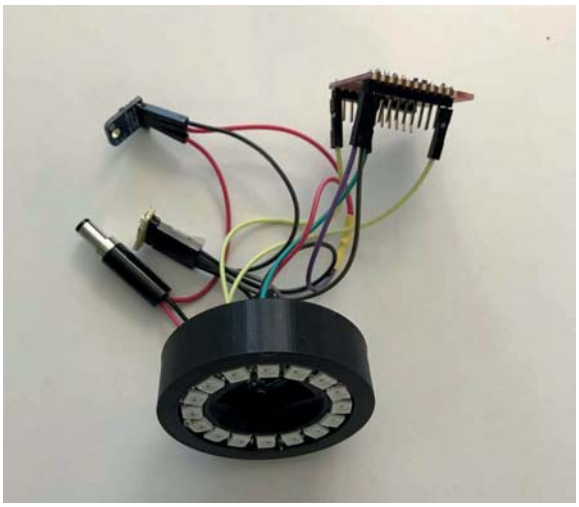


Abbildung 2: IoT-Button mit Lichtsensor, LED-Ring und ESP8266 mit WLAN-Modul

Der Lichtsensor kann auf einen schnellen Lichtwechsel reagieren, z.B. durch eine Wischbewegung über den Sensor, und ein Ereignis auslösen. Dieses Ereignis soll durch unterschiedliche Parteien verarbeitet werden können, um z.B. eine Lichtquelle einzuschalten, eine Präsentation zu starten, ein Objekt in Bewegung zu setzen oder dem Benutzer ein Feedback zu geben.

Als Hardware-Plattform verwenden wir im Schaufenster ein Raspberry Pi Model 3 [RPI3]. Mit dem darin integrierten WLAN-Modul kann diese Plattform auch die Funktion des Wireless Access Point übernehmen.

Der Lichtsensor als IoT-Button

Ausgehend von einem Lichtsensor (TSL2561) [TSL], einem LED-Ring, einem Stromwandler und einem Mikrocontroller mit integriertem WLAN-Modul haben wir einen IoT-Button entwickelt, welcher als Interaktionselement im Schaufenster verbaut wird (siehe Abb. 2). Der Mikrocontroller ESP8266 ist ein günstiger, weit verbreiteter 32-bit Mikrocontroller mit integriertem WLAN-Modul und genügend Speicher, um auch die IoT-Button-Applikation aufnehmen zu können [ESP]. Alle diese Komponenten sind in ein selbstkonstruiertes und mit einem 3D-Drucker gefertigtes Gehäuse eingebaut worden.

Der Mikrocontroller liest ununterbrochen die gemessenen Beleuchtungsstärken (Lux-Werte) des Lichtsensors aus. Eine erste Testphase hat gezeigt, dass die Detektion eines einfachen Intensitätssprunges des Lux-Wertes nicht robust genug ist, da sich die Lichtverhältnisse vor dem Schaufenster auch ohne Benutzerinteraktion rasch ändern können, denn das Schaufenster öffnet sich zu einem Platz hin mit einer stark wechselnden Sonneneinstrahlung. Eine Wischbewegung über den Lichtsensor hingegen führt zu einem detektierbaren Signal, welches als kurzer Ausschlag (Spike) sichtbar wird (siehe Abb. 3). Die Richtung des Ausschlages ist dabei irrelevant. Messungen mit einem ersten Sensor haben ergeben, dass die Dauer des Ausschlages im Bereich zwischen 0.1 und 1.5 Sekunden liegen kann. Für die Detektion werden die Lux-Werte ständig über die letzten hundert Messwerte gemittelt. Ausgehend vom Mittelwert m werden zwei Schwellwerte festgelegt: m +/- m_t. Sobald das Lichtsignal einen dieser beiden Schwellwerte über- bzw. unterschreitet wird eine Zeitmessung ausgelöst, welche wieder gestoppt wird, sobald der Lux-Wert den Mittelwert m wieder erreicht. Falls die Ausschlagdauer im gewünschten Bereich liegt, wird ein entsprechendes Ereignis ausgelöst und an eine zentrale Controller-Einheit übermittelt. Diese Controller-Einheit koordiniert alle Ereignisse im Schaufenster.

Sobald der IoT-Button ein Ereignis ausgelöst hat, gibt er über einen LED-Ring ein entsprechendes Feedback zurück. Dieser LED-Ring besitzt 24

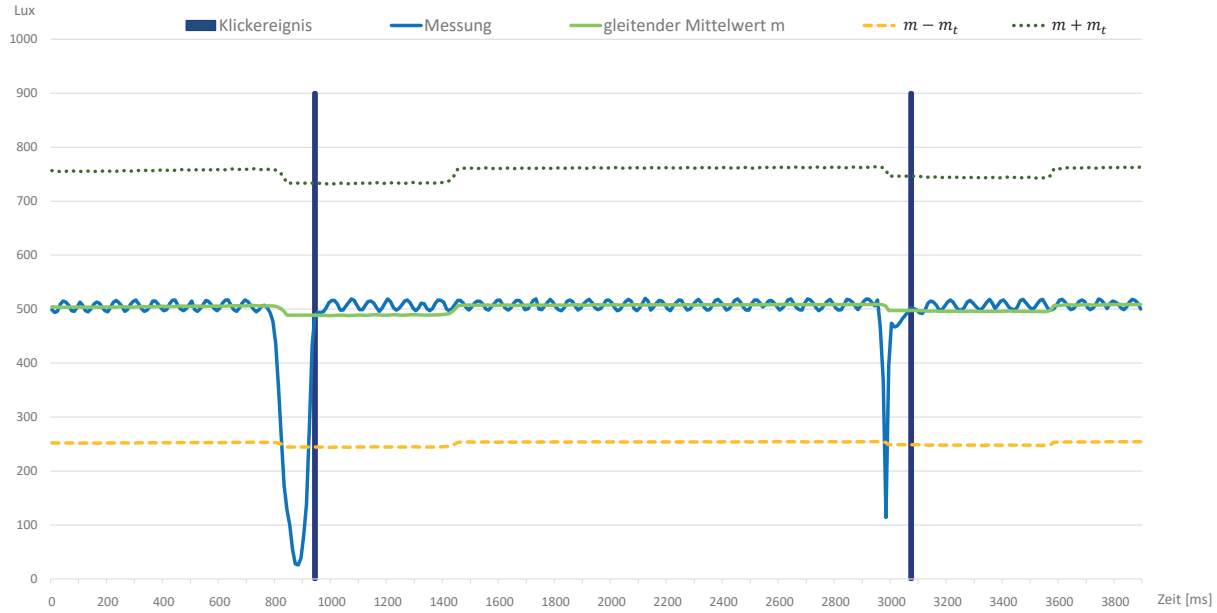


Abbildung 3: Lichtmessungen im Schaufenster. Die beiden senkrechten Balken zeigen ausgelöste Ereignisse an.

LEDs, die einzeln angesteuert werden können und mit denen unterschiedliche Feedbacks implementieren werden können. Der LED-Ring übernimmt aber noch eine weitere, nützliche Funktion: Er sorgt für eine konstante Lichtquelle, was vor allem bei schlechten Lichtverhältnissen am Abend und in der Nacht wichtig ist. Bei guten Lichtverhältnissen wird die Aktivierung des Sensors durch den Schattenwurf der Wischbewegung ausgelöst, bei schlechten Lichtverhältnissen hingegen durch die Reflektion des Lichtes des LED-Rings auf den Sensor.

MQTT als Message Broker

Auf ein Ereignis eines Lichtsensors sollen unterschiedliche Reaktionen gleichzeitig ausgelöst werden können. Zum Beispiel soll eine Lampe eingeschaltet werden, um ein entsprechendes Ausstellungsobjekt zu beleuchten, oder es soll eine Präsentation gestartet werden, die zum Ausstellungsobjekte weitere Informationen liefert. Zwischen Ereignisquelle und Ereignisempfänger besteht eine 1-n Beziehung. Diese Beziehung ist in der IoT-Welt typisch. Deshalb gibt es auch geeignete Kommunikationsinfrastrukturen, die über das Publish-Subscribe-Pattern diese 1-n Beziehung zwischen Ereignisquelle und Ereignisempfänger abbilden. Ein bekannter Vertreter ist MQTT [MQTT], ein offenes Nachrichtenprotokoll für Machine-to-Machine (M2M) Kommunikation, das seit 2013 über OASIS [OASIS] standardisiert wird.

MQTT ist eine leichtgewichtige Publish-Subscribe-Lösung mit der Komponente „Topic“ als Schnittstelle zwischen Publisher und Subscriber. Ein Topic lässt sich als schwarzes Brett mit eindeutiger Inventarnummer interpretieren, wobei sich alle anmelden müssen, die Informationen auf dem schwarzen Brett veröffentlichen wollen, ebenso diejenigen, die Nachrichten lesen wollen. Die Topics werden als hierarchische Baumstruktur aufgebaut, um eine eindeutige Identifikation zu ermöglichen.

Die zentrale Komponente von MQTT ist ein sogenannter MQTT-Broker, der die Topics inklusive der darin enthaltenen Nachrichten verwaltet, sowie den Zugriff auf die Topics regelt. Für Datensicherheit sind entsprechende Mechanismen vorhanden. Als MQTT-Broker wird Moquette eingesetzt [Moq]. Dies ist eine in Java implementierte Open Source Lösung.

IoT-Button mit MQTT Technologie

Die hierarchische Struktur der Topics für das Schaufenster ist einfach gehalten. Für jeden IoT-Button existiert ein entsprechendes Topic „sensor/i“, wobei i die eindeutige Sensornummer ist. Jeder IoT-Button publiziert seine Button-Ereignisse auf seinem eigenen Topic in Form einer MQTT-Message. Aktoren, welche sich für IoT-Button-Ereignisse interessieren, schreiben sich bei den entsprechenden Topics als Empfänger ein. Damit ein IoT-Button selber ein Feedback für seinen LED-Ring erhalten kann, muss er sich auf seinem eigenen Topic auch als Empfänger einschreiben.

Andere Subscriber/Aktoren in unserem System sind die Lampen für die Beleuchtung der Ausstellungsgegenstände und der Informationsmonitor. Für die Beleuchtung verwenden wir Philips Hue-Lampen, da sie über das drahtlose Kommu-

nikationsprotokoll Zigbee kontrolliert werden können und viele Funktionalitäten zur Verfügung stellen, welche die Ausstellung bereichern und den heutigen Stand der IoT-Lampen darstellen [Hue, Zig]. Eine Philips Hue-Lampe verfügt über ein REST-API, welches über eine HTTP-Schnittstelle angesprochen werden kann. Diese Schnittstelle kann auch aus MQTT bedient werden, indem ein entsprechender Light-Service implementiert wird, der die Übersetzung zwischen MQTT und HTTP übernimmt. Der Light-Service meldet sich beim MQTT-Broker als Subscriber an und sobald er ein MQTT-Ereignis empfängt, sendet er einen entsprechenden HTTP-Request zur entsprechenden Philips Hue-Bridge, welche dann über Zigbee die dazugehörige Lampe schaltet. Die Einbindung von Hue-Lampen zeigt schön auf, wie bereits bestehende Systeme einfach in das Ökosystems des Schaufensters aufgenommen werden können.

Wie schon erwähnt, befindet sich im Schaufenster auch ein Informationsbildschirm mit zugehörigem Rechner (Mini-PC-System NC2000XA [XPC]). Darauf können Zusatzinformationen zur Ausstellung und den Objekten gezielt angezeigt werden. Dieser Informationsbildschirm ist ein weiterer Aktor. Er reagiert auf die Ereignisse der Lichtsensoren, um die entsprechende Diashow vorzubereiten und zu starten. Dazu wird die Software-Bibliothek Pygame verwendet [Pyg]. Pygame ist eine kleine Python-Bibliothek, welche den VLC-Player über den Simple DirectMedia Layer (SDL) steuern kann [SDL]. Damit ohne nennenswerten Aufwand zwischen verschiedenen Diashows umgeschaltet werden kann, werden die Bilder und Videos in ein entsprechendes Directory hochgeladen und dort automatisch zu einem Film zusammengefügt und an das Format des Informationsbildschirmes angepasst. Die daraus entstandenen Videos werden dann mit einem VLC-Wrapper im VLC-Player abgespielt.

Flexible Konfiguration einer Ausstellung

Für die Konfiguration der IoT-Buttons und Aktoren existieren drei weitere Topics: „discover“, „available“ und „update“. Die Aufteilung in drei Topics spart Ressourcen und vereinfacht die Logik der Nachrichtenbehandlung in den IoT-Buttons. Die Topics „discover“ und „update“ funktionieren ähnlich wie ein Broadcast: Alle Aktoren und Sensoren im System müssen sich zwingend bei diesen Topics einschreiben. Wenn das Schaufenster konfiguriert werden soll wird über das Topic „discover“ eine Nachricht zu jedem Gerät gesendet. Die Geräte melden sich daraufhin auf dem Topic „available“ unter Angabe ihrer ID.

Anschliessend an diese Initialisierungsphase kann das Schaufenster konfiguriert werden. Zur Konfiguration steht die in Abbildung 5 gezeigte Webapplikation zur Verfügung. Dies Applikation stellt alle im System bekannten Geräte grafisch dar. Aktoren und Sensoren lassen sich einfach zu dem entsprechenden Topic „sensor/i“ zuordnen. Sobald die manuelle Konfiguration abgeschlossen ist, generiert die Applikation eine Konfigurations-

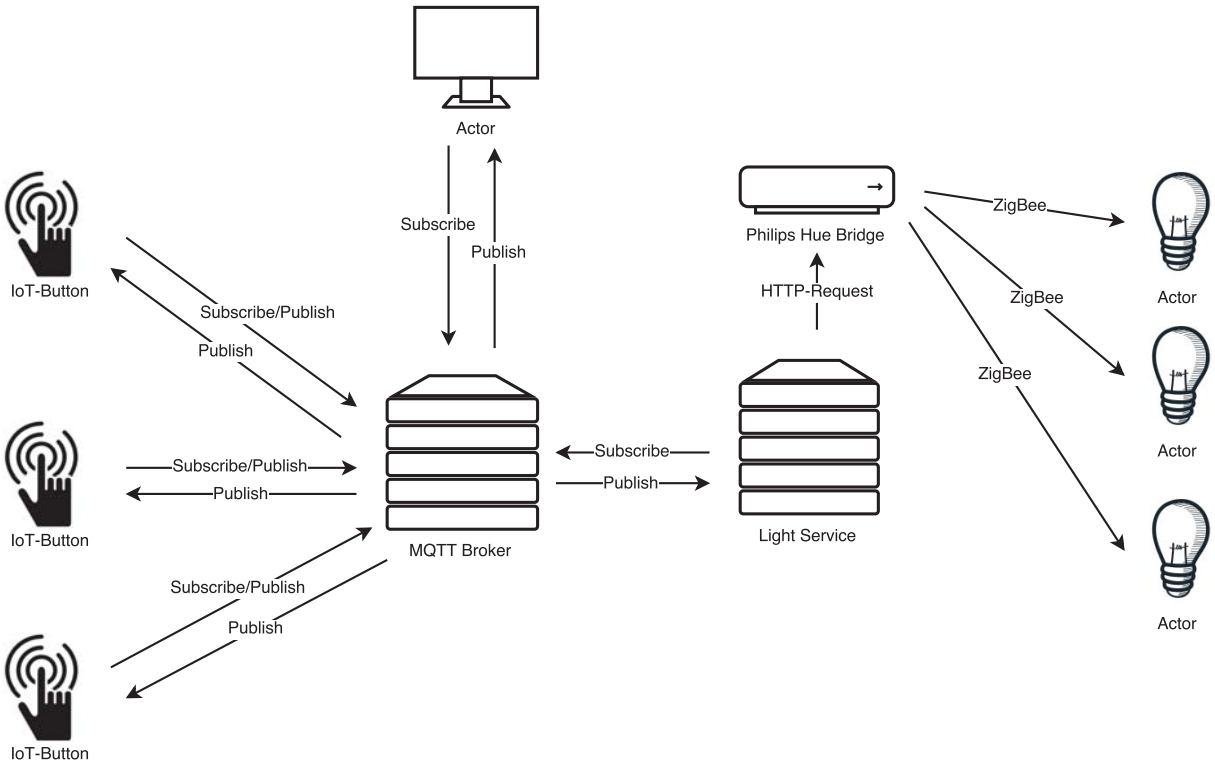


Abbildung 4: Konkreter Einsatz von MQTT mit IoT-Button als Publisher/Subscriber und Lampe/Bildschirm als Subscriber

Abbildung 5: Web-App zur Konfiguration: Lampen, Relais und Geräte können mit Schaltern (IoT-Button) verknüpft werden

datei im JSON-Format, welche dann unter dem Topic „update“ veröffentlicht wird, so dass jedes Gerät im System über das neue Konfigurationslayout informiert ist.

Die Webapplikation selber ist mit Node.js implementiert unter unterstützt ein responsives Webdesign, so dass die Benutzerin die Konfiguration mit einem Tablet, Laptop oder sogar einem Smartphone direkt vor dem Schaufenster vornehmen kann [NJS].

Zusammenfassung

Das Schaufenster bietet eine Plattform, um interaktive Ausstellungen präsentieren zu können. Die Interaktionselemente werden mit selbst entwickelten IoT-Buttons realisiert, die am Schaufenster angeklebt sind und die auf Änderungen im Lichtsignal reagieren. Die eigentlichen Ausstellungsgegenstände werden auf verstellbaren Tablaren angeordnet, durch IoT-Buttons gesteuert und mittels drahtlos angesteuerten Lampen beleuchtet. Über den Informationsbildschirm werden Bilder und Videos über die aktuelle Ausstellung präsentiert.

Eine selbst angefertigte Rahmenkonstruktion trägt die Tablare, die dynamisch in der Höhe und in der Länge angeordnet werden können. Die Kabelkanäle für die Stromversorgung verlaufen im Rahmen selbst und sind für den Betrachter unsichtbar.

Die Kommunikation baut auf MQTT auf. Detektierte Ereignisse der IoT-Buttons werden publiziert und an die entsprechenden Empfänger weitergeleitet, die anschliessend ihre Aktionen ausführen. Für Aktoren, welche nicht direkt ins System eingebunden werden können, weil ihnen die nötige MQTT-Schnittstelle fehlt, wie z.B. die Philips Hue-Lampen, wird ein zusätzlicher Dienst angeboten, welcher die MQTT-Nachrichten empfängt, umwandelt und über eine entsprechende Schnittstelle an die Aktoren weiterleitet.

Referenzen

[ESP] Espressif ESP32 – A Different IoT Power & Performance: https://espressif.com/en/products/hardware/esp32/overview

[Hue] Philips hue: https://www2.meethue.com/de-de

[Mak] Maker: https://de.wikipedia.org/wiki/Maker

[MakStu] Maker Studio: https://web.fhnw.ch/plattformen/makerstudio

[Moq] Java MQTT lightweight broker: https://github.com/andsel/moquette

[MQTT] MQTT: http://mqtt.org

[NJS] Node.js: https://nodejs.org/en

[OASIS] OASIS: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=mqtt

[Pyg] Pygame: https://www.pygame.org/news

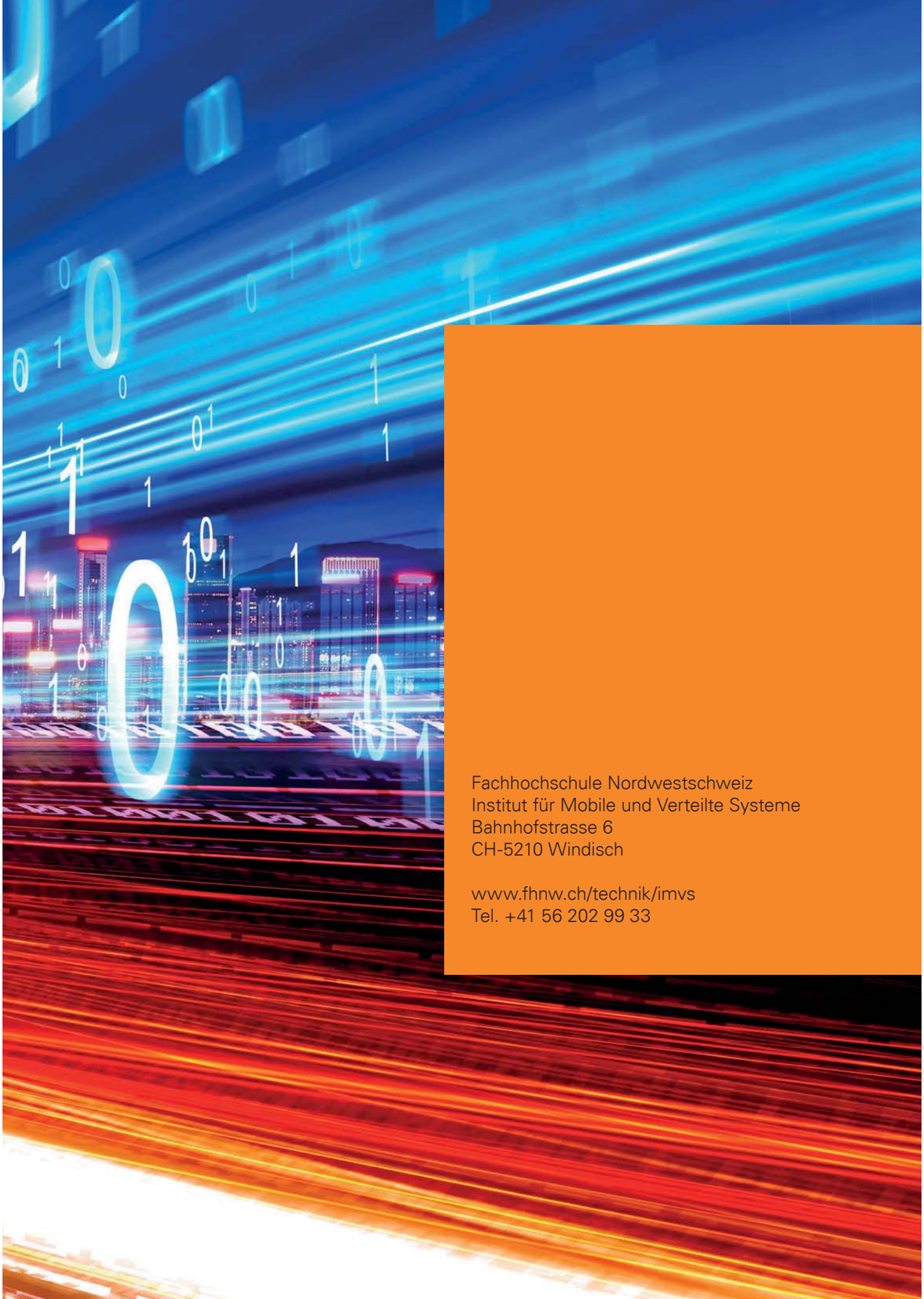
[RPI3] Einplatinencomputer, Website Raspberry Pi: https://www.raspberrypi.org

[SDL] Simple DirectMedia Layer: https://de.wikipedia.org/wiki/Simple_DirectMedia_Layer

[TSL] Adafruit TSL2561: https://www.adafruit.com/product/439

[XPC] Shuttle XPC nano NC2000XA: http://www.shuttle.eu/products/nano/nc2000xa

[Zig] Zigbee: http://www.zigbee.org



Fachhochschule Nordwestschweiz
Institut für Mobile und Verteilte Systeme
Bahnhofstrasse 6
CH-5210 Windisch

www.fhnw.ch/technik/imvs
Tel. +41 56 202 99 33